**EXHIBIT E-1**
Defendant's First Amended Invalidity Contentions
*Orckit Corporation v. Cisco Systems, Inc.*, 2:22-cv-00276-JRG-RSP

---

**Chart for U.S. Patent 6,680,904 ("the '904 Patent")**
**35 U.S.C. § 103**

In this chart, "The Reference" refers to each of the following references and/or systems:

- Cisco Catalyst 2900 Series XL and 3500 Series XL Switches (collectively, the "Catalyst XL Switches")
- BayStack 450 Series Switches (the "BayStack 450 Switches")
- TRENDnet Stackable Hubs ("TRENDnet Stackable Hubs")
- U.S. Patent No. 6,314,102 to Czerwic ("Czerwiec")
- U.S. Patent No. 6,092,214 to Quoc ("Quoc")
- PCT Application No. WO 91/14324 to Vink ("Vink")
- U.S. Patent No. 6,600,727 to Mackay ("Mackay")
- U.S. Patent No. 6,663,499 to Dowling ("Dowling")
- U.S. Patent No. 5,953,318 to Nattkemper ("Nattkemper")

The following additional references are identified individually:

- U.S. Patent No. 5,313,456 to Sugawara ("Sugawara")
- U.S. Patent No. 6,654,796 to Slater et al. ("Slater '796")
- U.S. Patent No. 6,895,433 to Slater & Chennapragada ("Slater '433")
- U.S. Patent No. 6,917,626 to Duvvury ("Duvvury '626")
- U.S. Patent No. 7,545,820 to Duvvury ("Duuvury '820")
- U.S. Patent No. 6,269,452 to Daruwalla et al. ("Daruwalla")
- U.S. Patent No. 6,853,623 to Nederveen & King ("Nederveen")
- U.S. Patent No. 6,952,421 to Slater ("Slater '421")
- U.S. Patent No. 6,463,065 to Petersen et al. ("Petersen")

- *Cisco Introduces Next-Generation Stacking with New Catalyst 3500 Series XL*, CISCO: THE NEWSROOM, May 24, 1999, https://newsroom.cisco.com/c/r/newsroom/en/us/a/y1999/m05/cisco-introduces-next-generation-stacking-with-new-catalyst-3500-series-xl.html ("Cisco Catalyst Press Release")

As shown in the chart below, all Asserted Claims of the '904 Patent are invalid under 35 U.S.C. § 103 because The Reference renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions.

Motivations to combine include at least the similarity in subject matter between the references to the extent they concern methods of stackable switching. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

These invalidity contentions are based on Defendant's present understanding of the Asserted Claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| 1[preamble] | Network access apparatus, comprising: | The Reference discloses network access apparatus, comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |
| 1[a] | first and second master units, each comprising a physical interface to a packet-switched network; | The Reference discloses first and second master units, each comprising a physical interface to a packet-switched network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626<br>• Duuvury '820 |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | **<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2. |

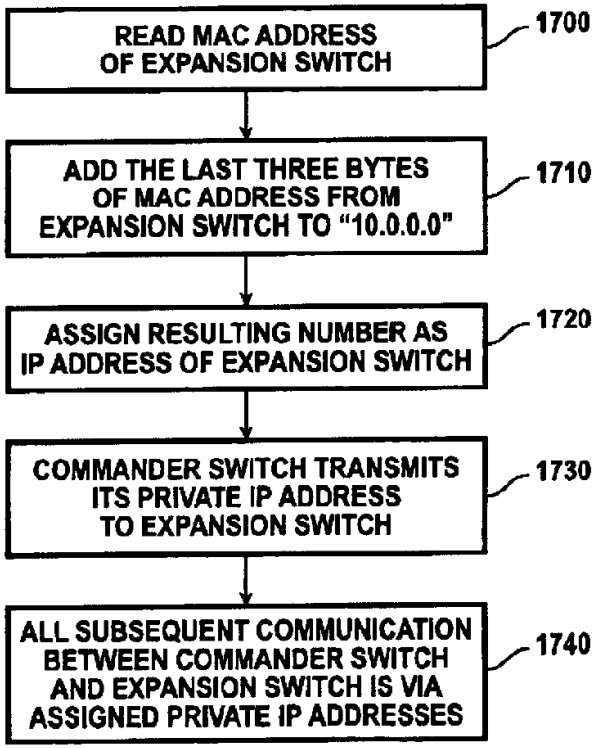
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

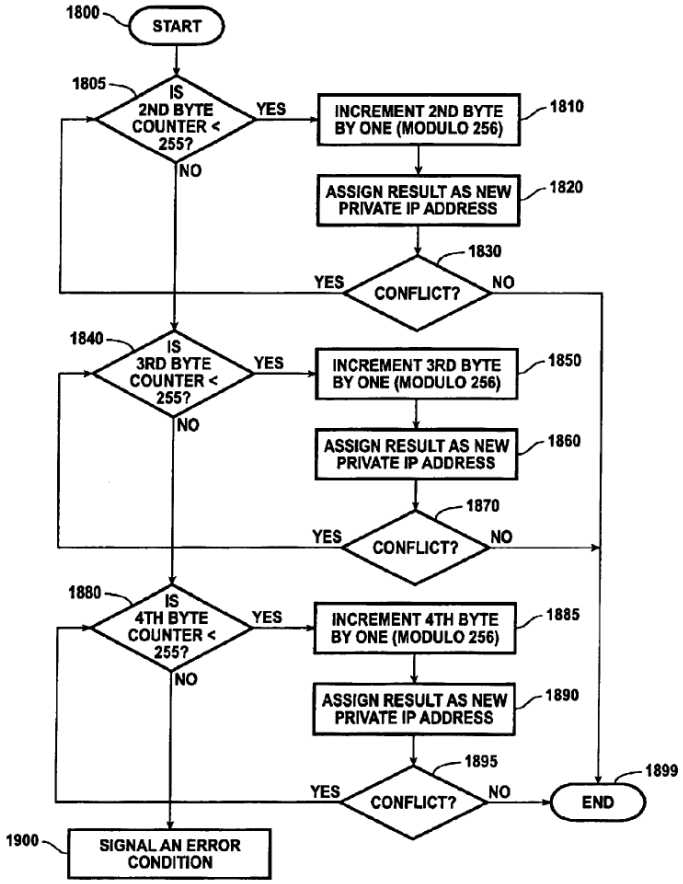
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:[1]<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP |

[1] Duvvury '626 is the parent to the CON Duvvury '820 and contains a substantially identical specification and reads on the Asserted Claims of the '904 patent for substantially the same reasons as Duvvury '626.

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract.<br><br>"If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br><br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, |

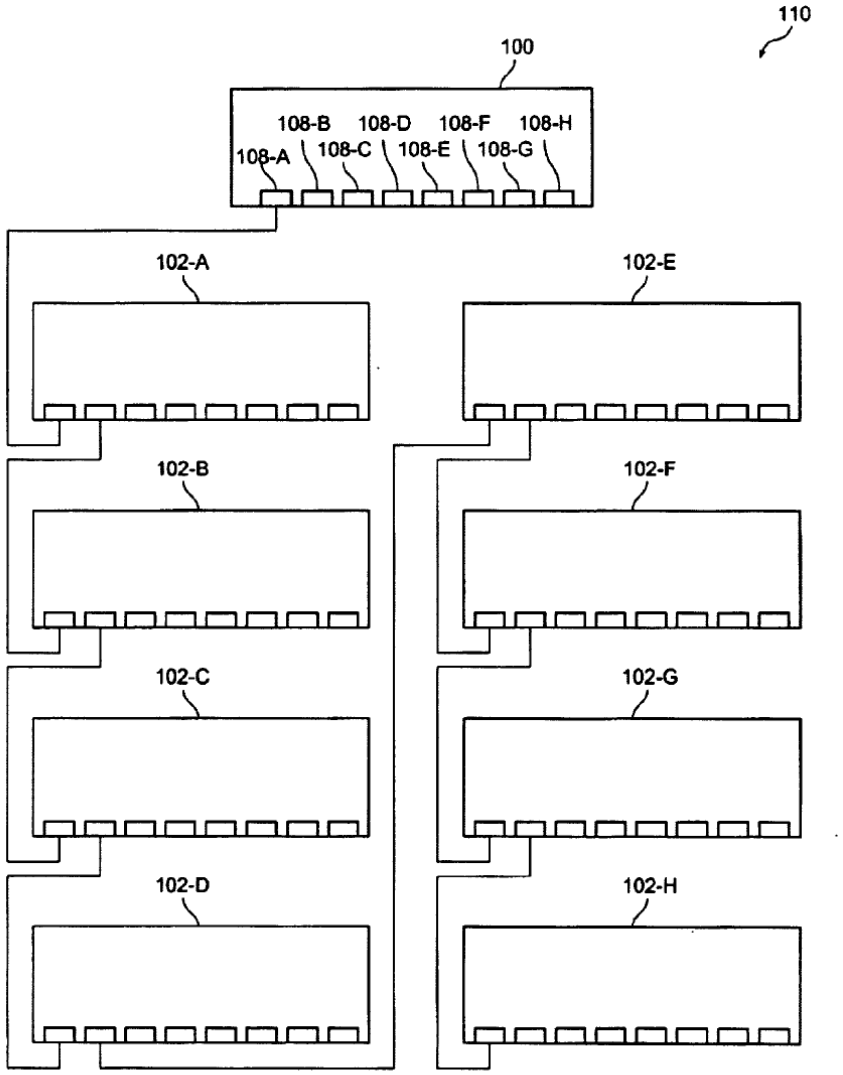
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16.   *FIG. 18* Duvvury '626, FIG. 18. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:[2]<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

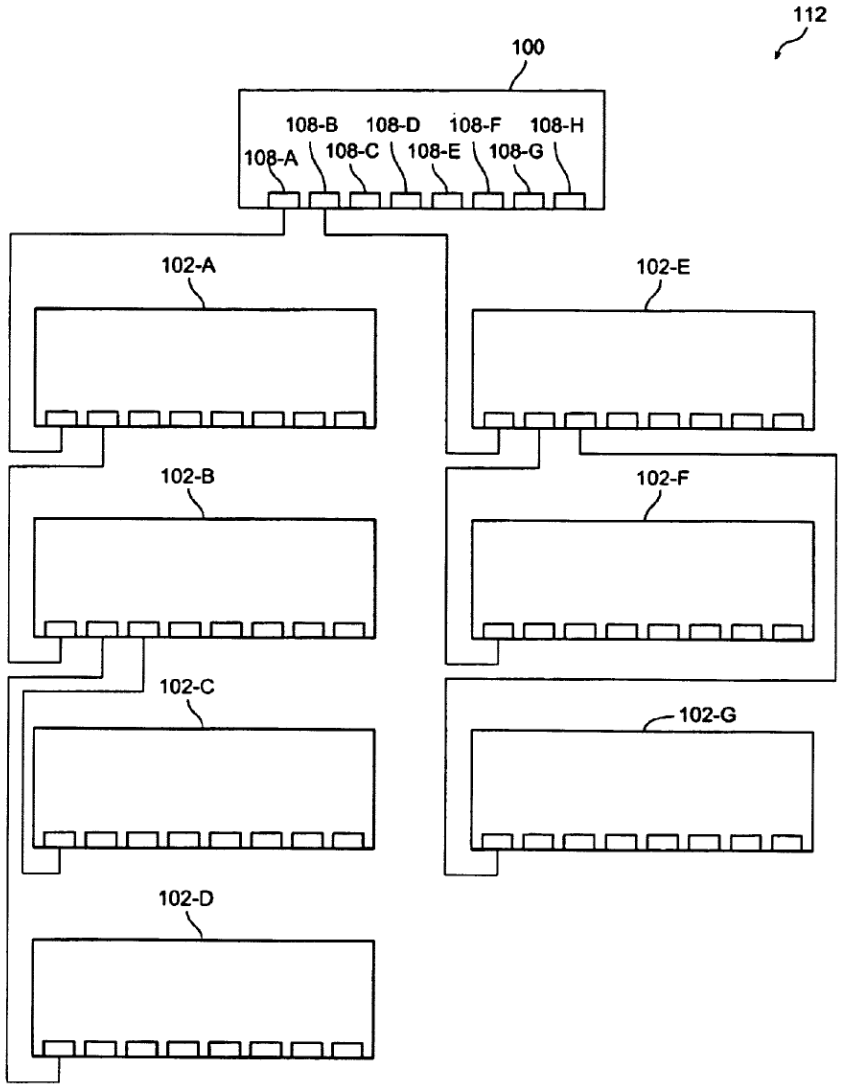
[2] Slater '433 is the parent to the CON Slater '796 and contains a substantially identical specification and reads on the Asserted Claims of the '904 patent for substantially the same reasons as Slater '796.

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 1[b] | a plurality of slave units, | The Reference discloses a plurality of slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626 |

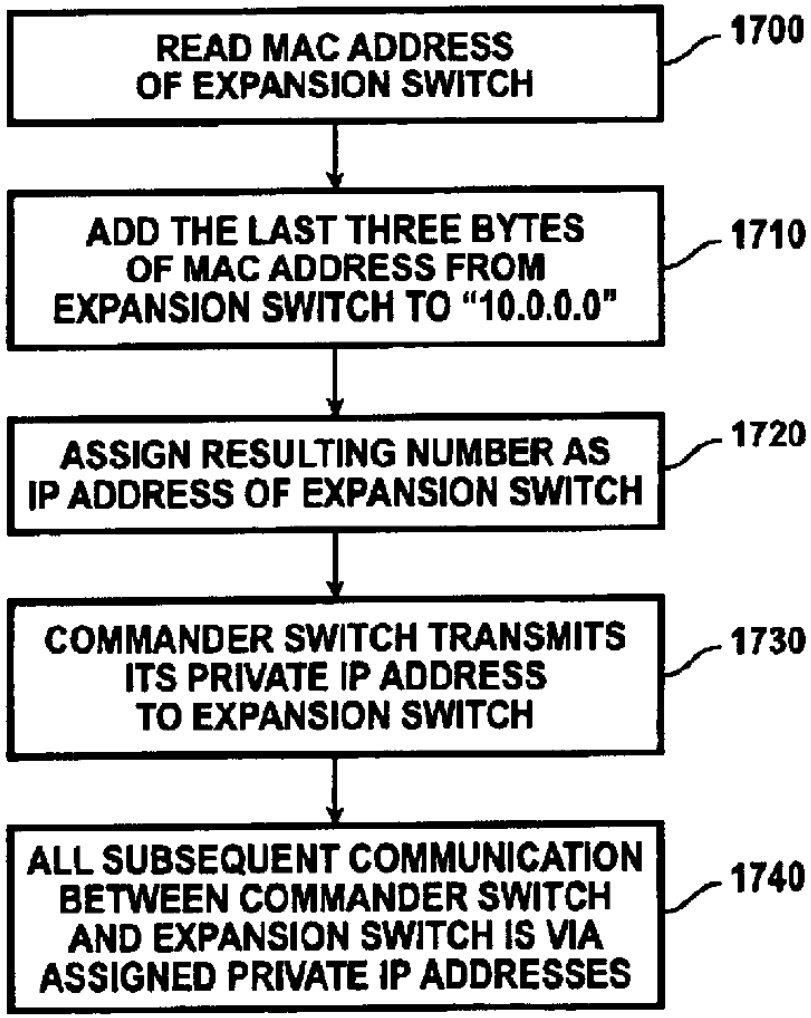
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | • Duuvury '820<br><br>**<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

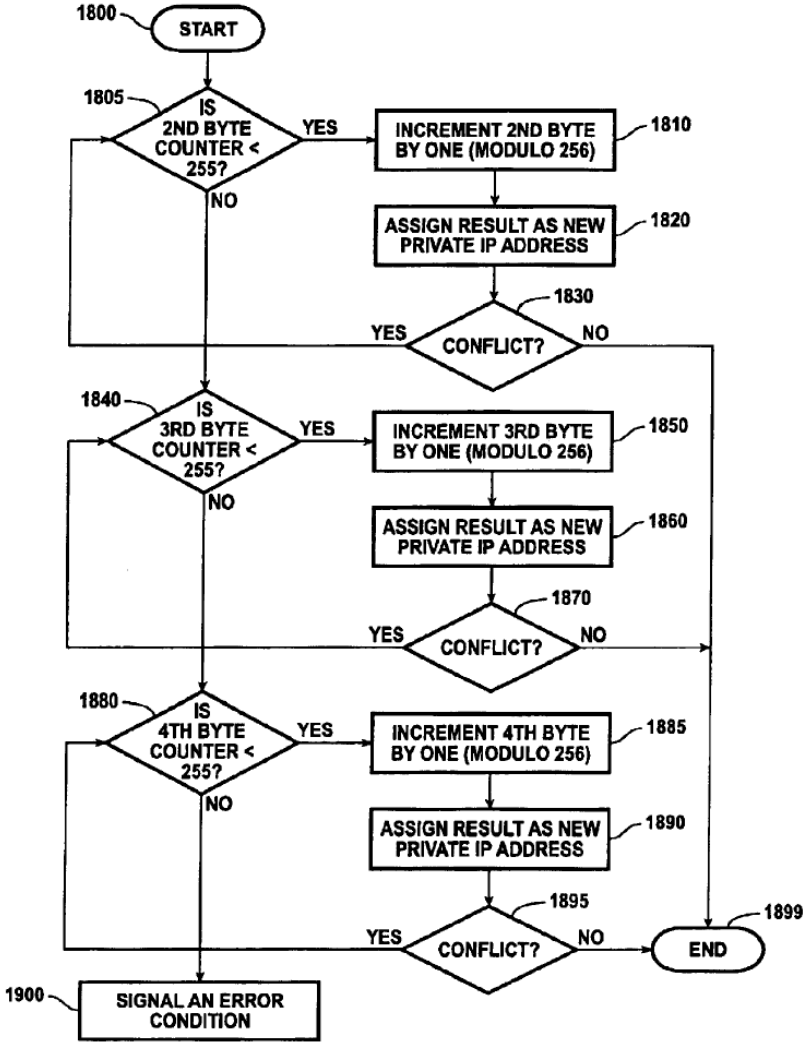
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17. |

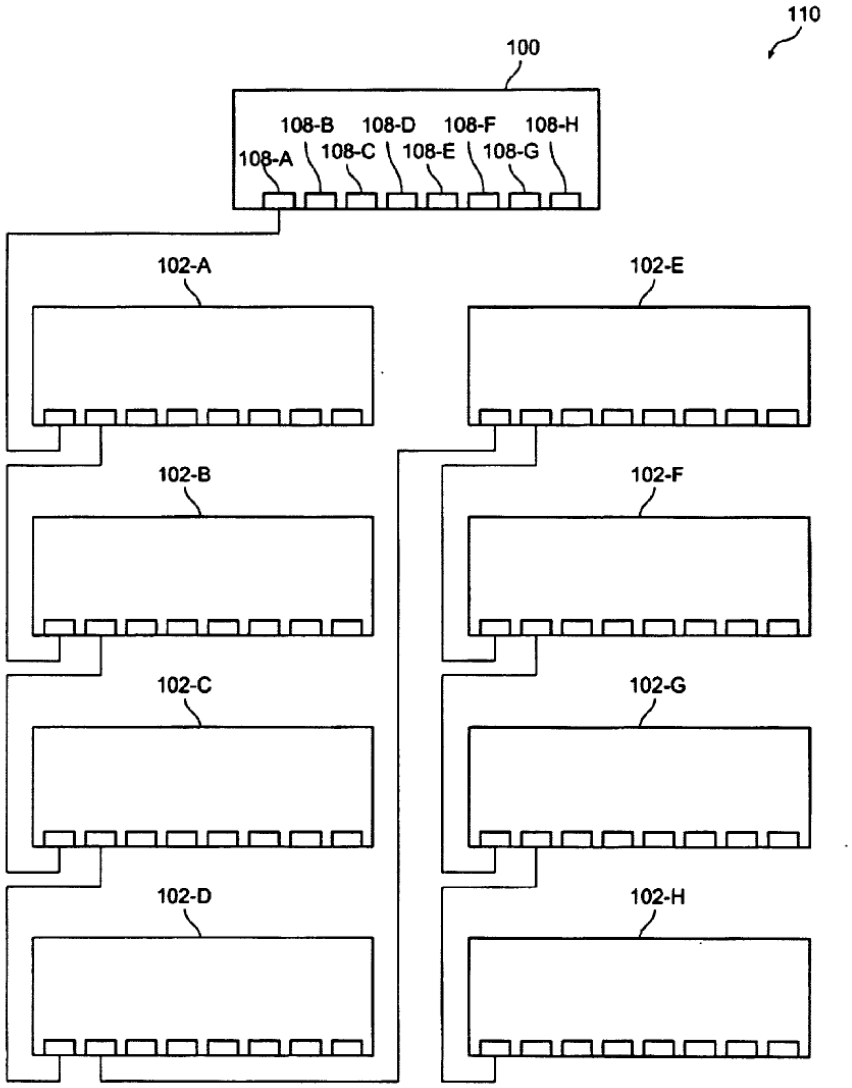
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

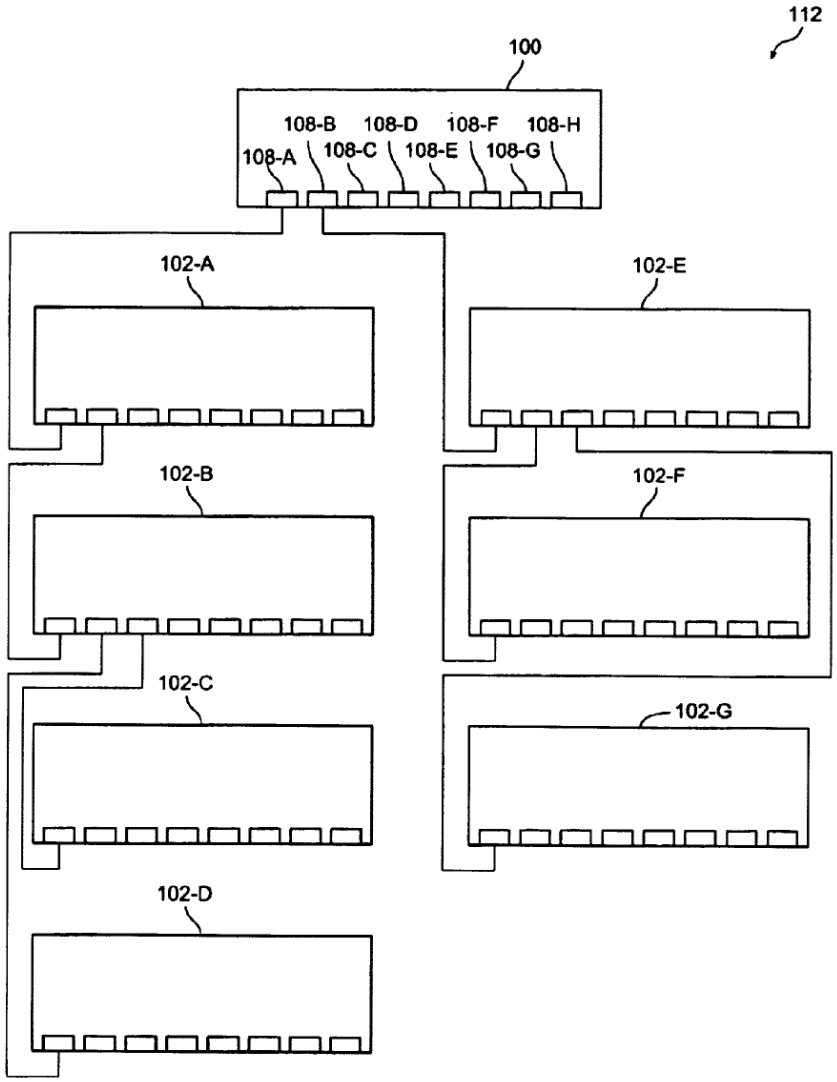
| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 1[c] | each slave unit comprising one or more ports to respective subscriber lines; and | The Reference discloses each slave unit comprising one or more ports to respective subscriber lines.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |
| 1[d] | a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by | The Reference discloses a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | the physical interface lines therebetween, | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling and Sugawara.<br><br>Below are examples of such references.<br><br>**<u>Sugawara discloses</u>**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

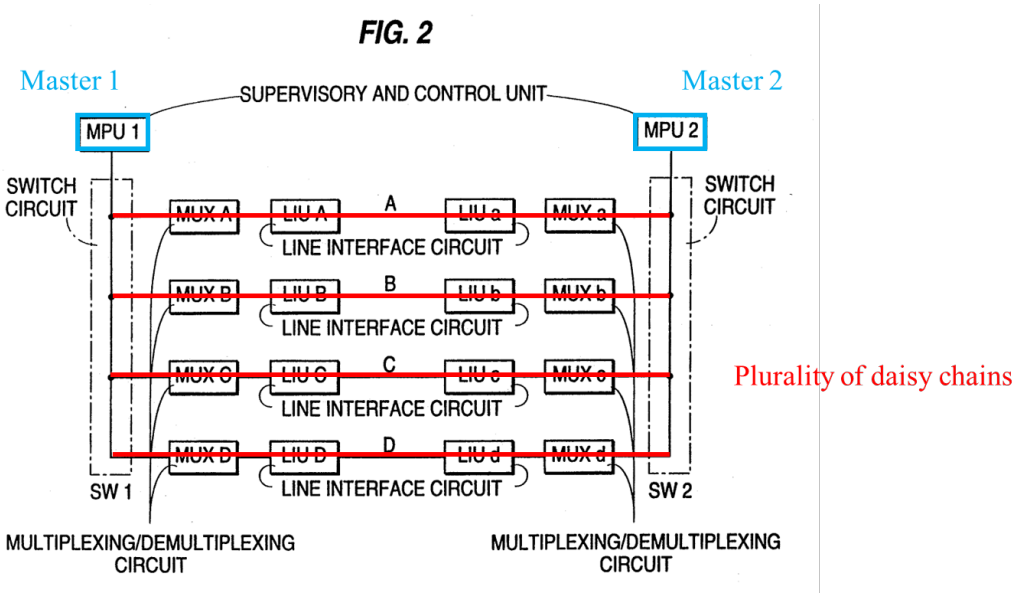



FIG. 2 (annotated).

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| | | Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

| No. | '904 Patent Claim 1 | The Reference |
|---|---|---|
| 1[e] | each daisy chain comprising at least a first slave unit connected one of the physical interface lines to the first master unit, a second slave unit connected to the first slave unit but not to the first or second master unit, and a last slave unit connected by another of the physical interface lines to the second master unit. | The Reference discloses each daisy chain comprising at least a first slave unit connected one of the physical interface lines to the first master unit, a second slave unit connected to the first slave unit but not to the first or second master unit, and a last slave unit connected by another of the physical interface lines to the second master unit.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 2 | The Reference |
|---|---|---|
| 2 | Apparatus according to claim 1, wherein the network comprises an asynchronous transfer mode (ATM) network. | The Reference discloses Apparatus according to claim 1, wherein the network comprises an asynchronous transfer mode (ATM) network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 3 | The Reference |
|---|---|---|
| 3 | Apparatus according to claim 1, wherein the network comprises an Internet protocol (IP) network. | The Reference discloses apparatus according to claim 1, wherein the network comprises an Internet protocol (IP) network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| 4[preamble] | Network access apparatus, comprising: | The Reference discloses network access apparatus, comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[preamble].<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626 |

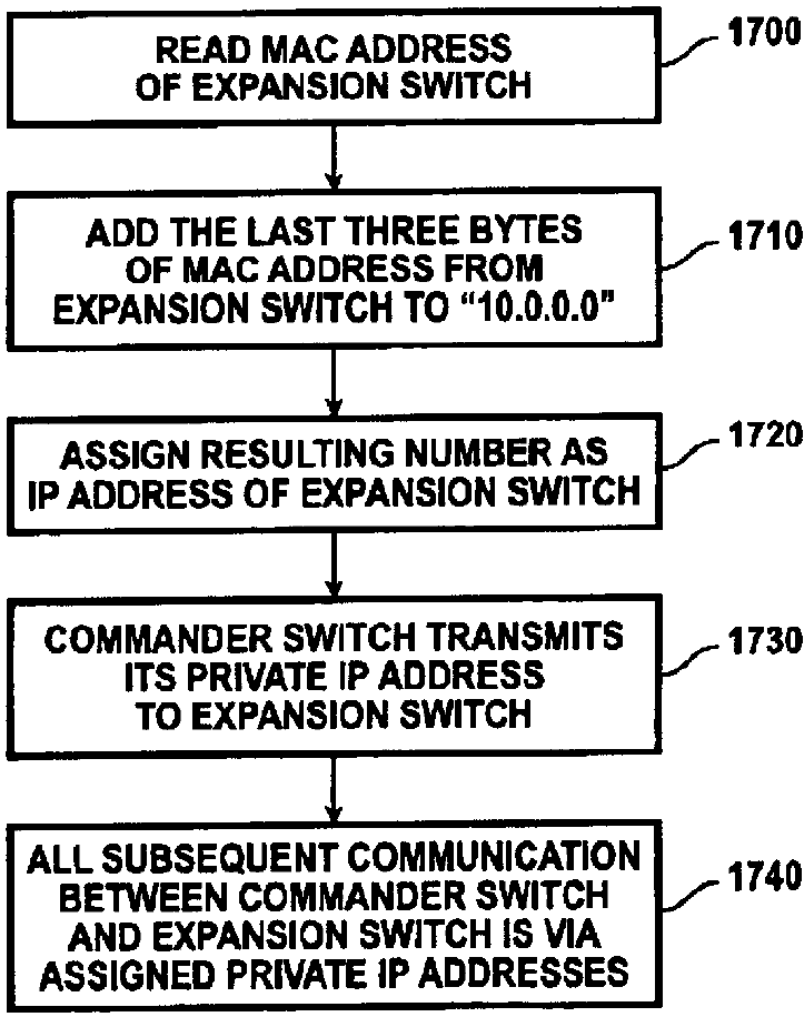
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | • Duuvury '820<br><br>**<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

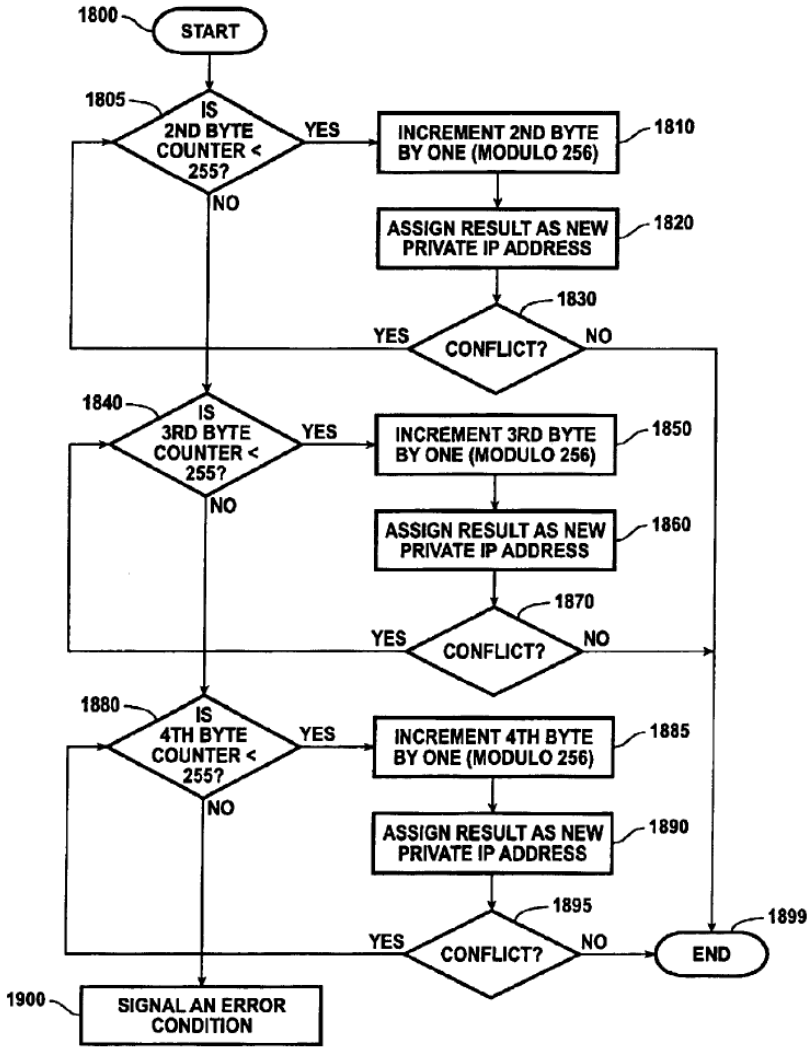
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17. |

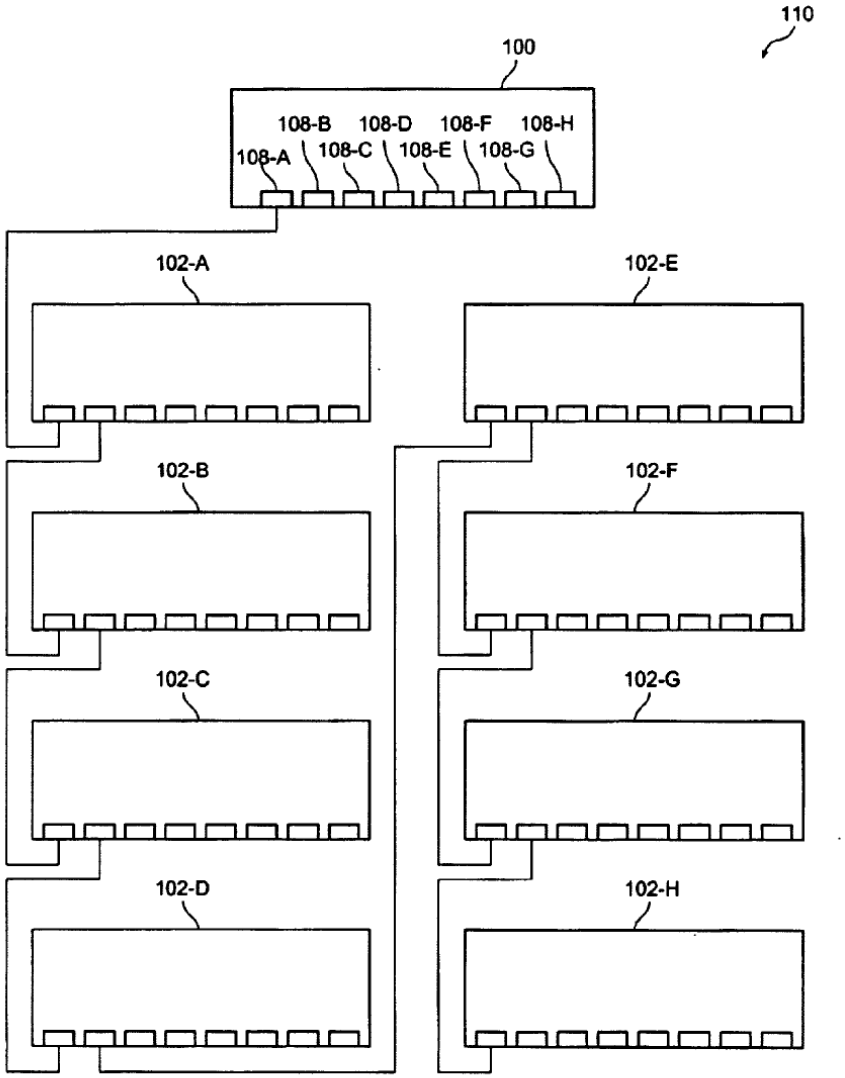
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

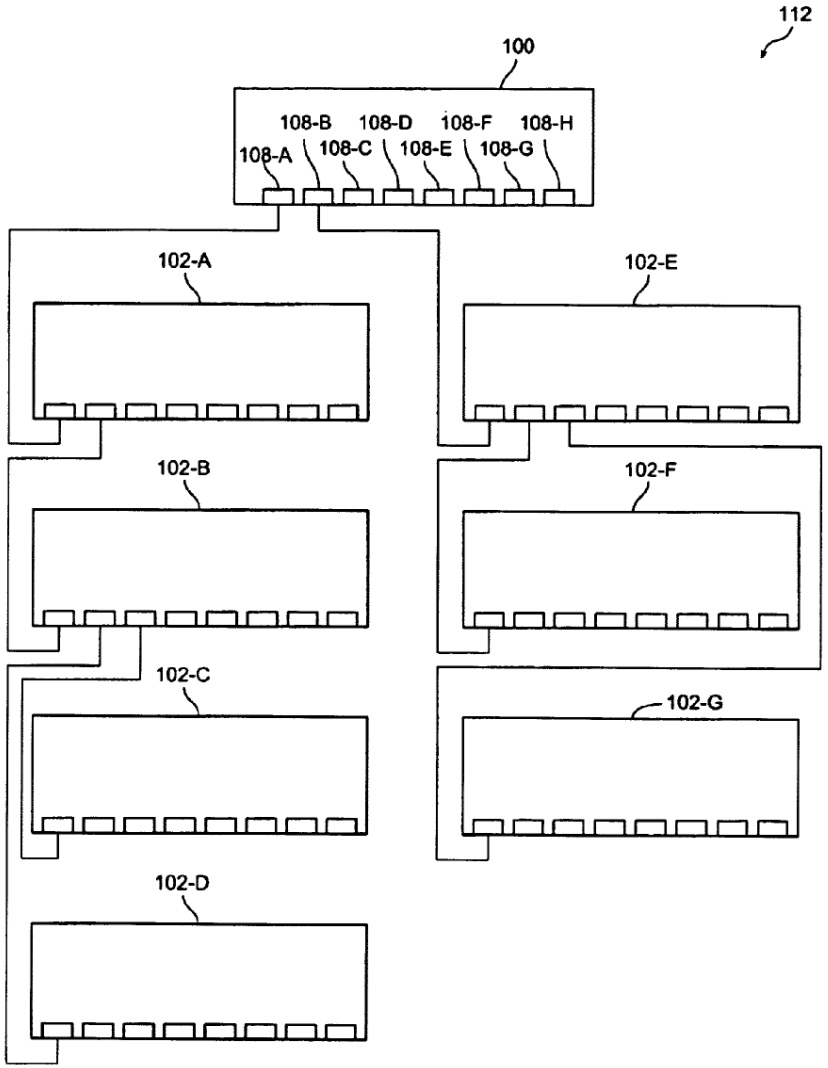
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

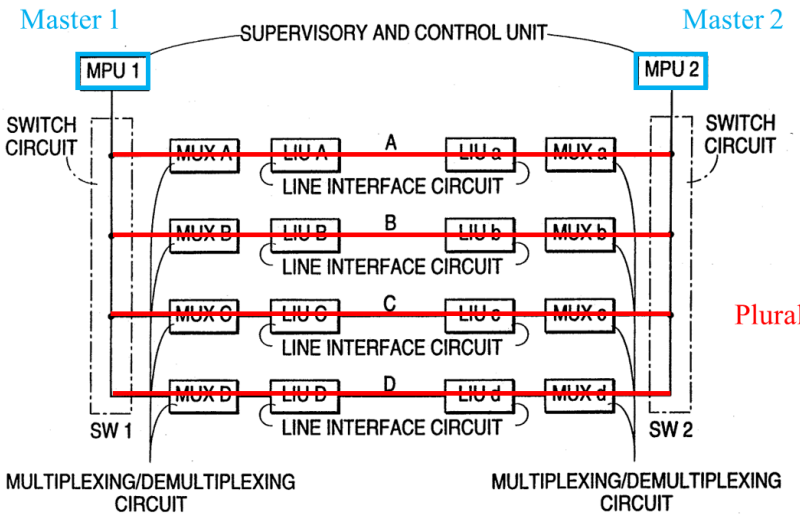
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 4[a] | first and second master units, each comprising a physical interface to a packet-switched network; | The Reference discloses first and second master units, each comprising a physical interface to a packet-switched network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[a]. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| 4[b] | a plurality of slave units, | The Reference discloses a plurality of slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[b].<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626<br>• Duuvury '820<br><br>Below are examples of such references.<br><br>**Cisco Catalyst Press Release discloses**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2.<br><br>"With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3. |

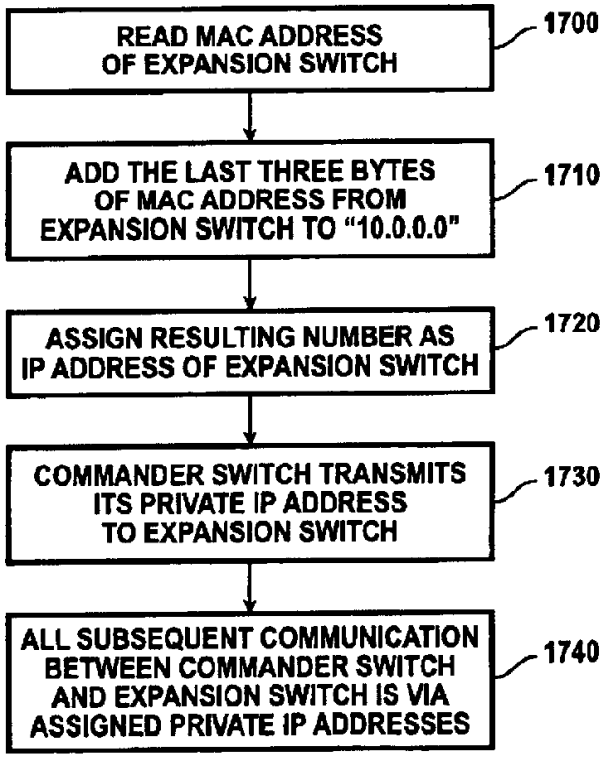
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the |

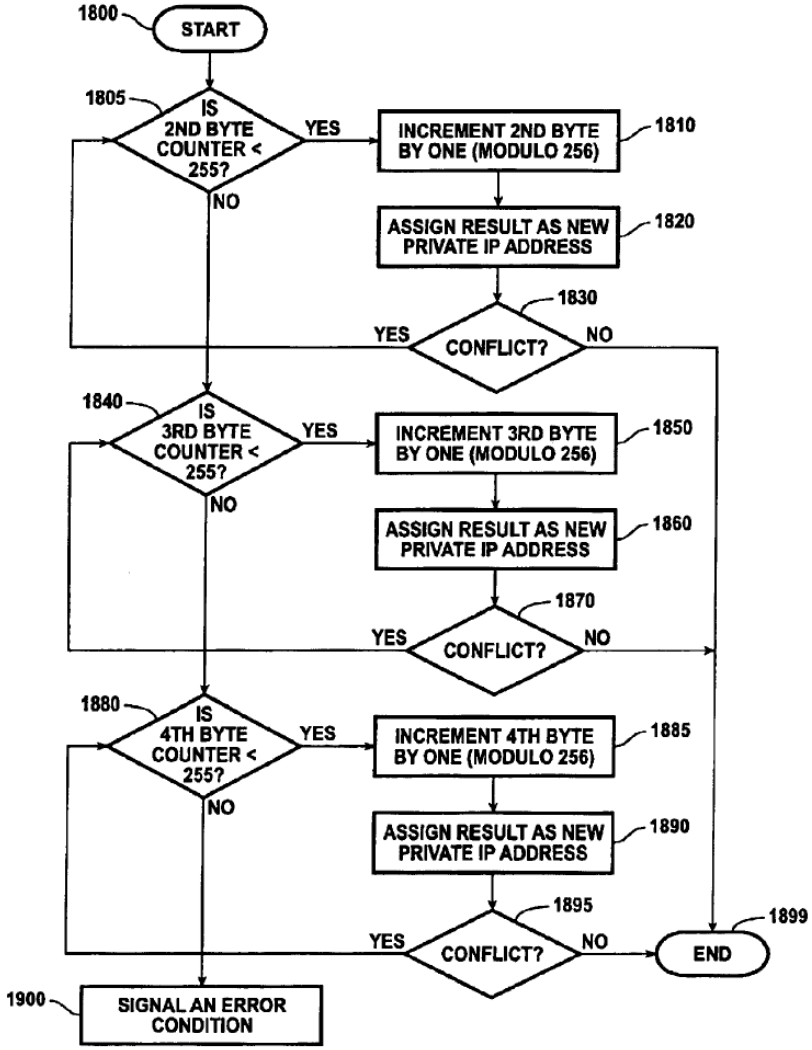
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided.").<br><br>*FIG. 2*<br>  <br><br>FIG. 2 (annotated). |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | **Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract.<br><br>"If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31.<br>  <br><br>*FIG. 17*<br>Duvvury '626, FIG. 17. |

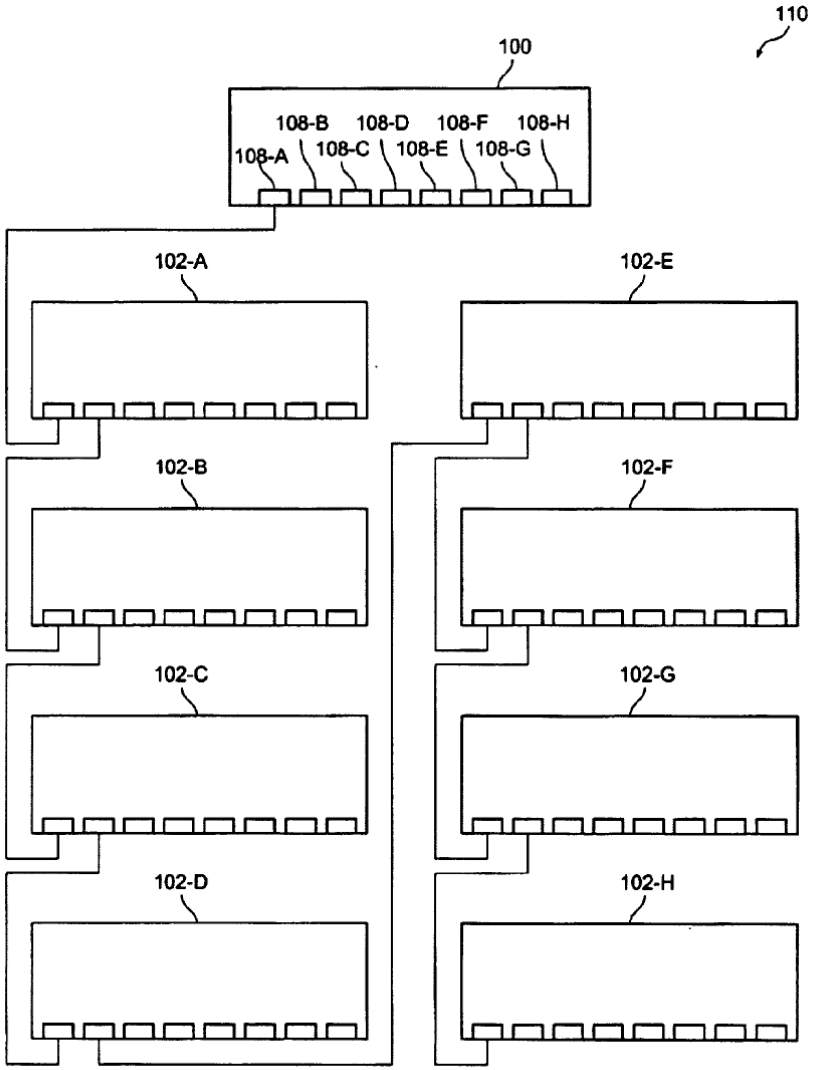
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

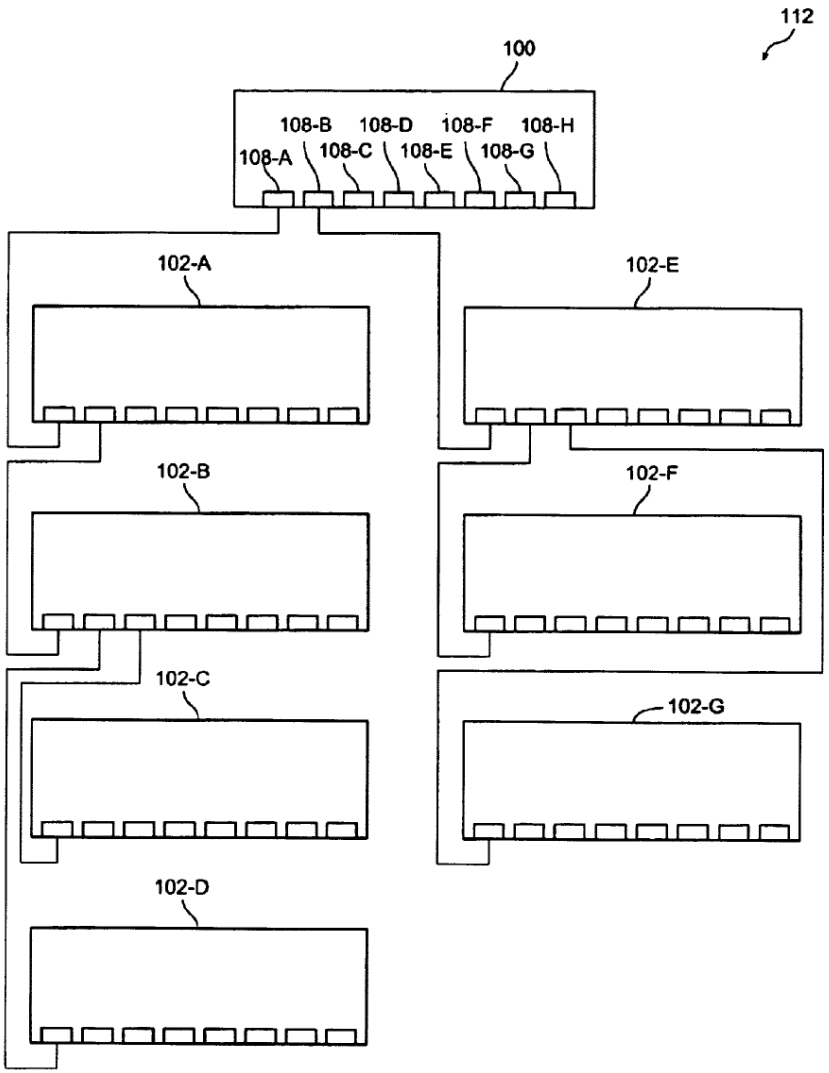
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10.<br><br>"Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

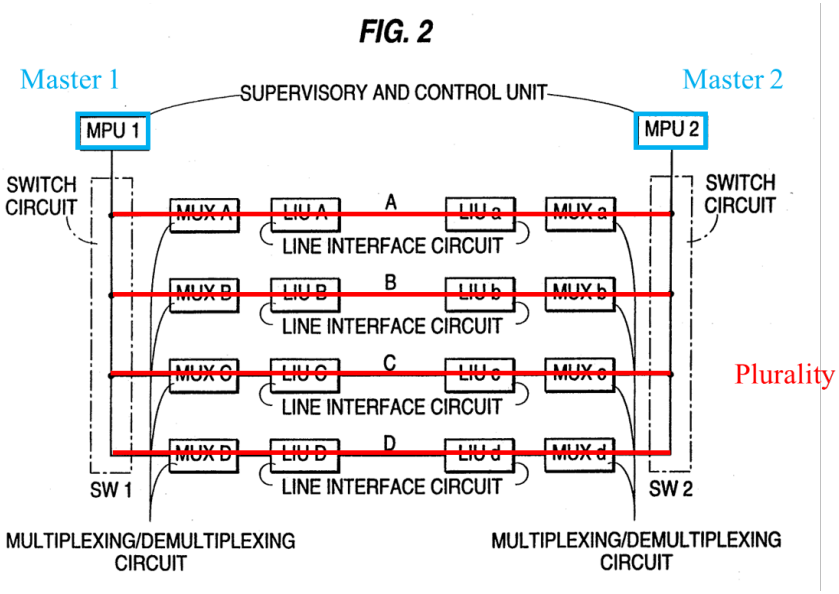
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 4[c] | each slave unit comprising one or more ports to respective subscriber lines; and | The Reference discloses each slave unit comprising one or more ports to respective subscriber lines.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[c]. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| 4[d] | a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween, | The Reference discloses a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling and Sugawara.<br><br>*See supra* at 1[d]. |
| 4[e] | each daisy chain comprising at least a first slave unit connected by one of the physical interface lines to the first master unit and a last slave unit connected by another of the physical interface lines to the second master unit | The Reference discloses each daisy chain comprising at least a first slave unit connected by one of the physical interface lines to the first master unit and a last slave unit connected by another of the physical interface lines to the second master unit.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[e].<br><br>Below are examples of such references. |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | **Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided.").<br>  <br><br>FIG. 2 (annotation added)<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl |

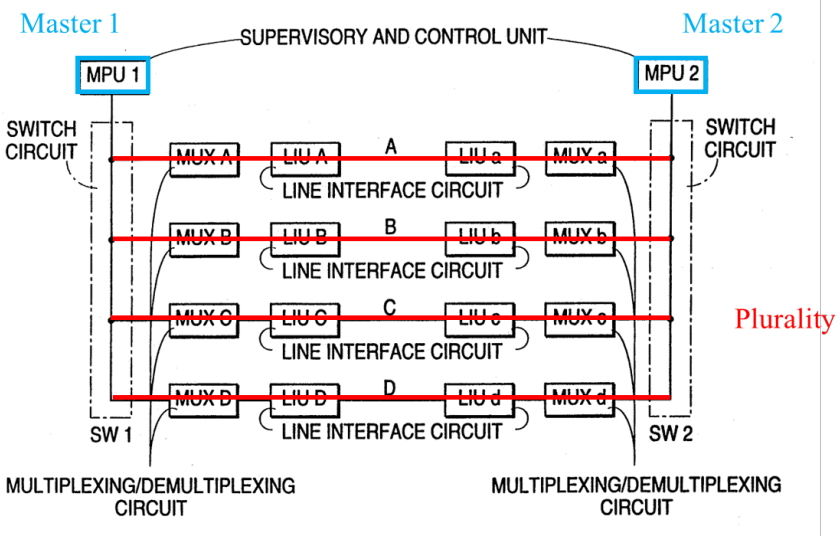
| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| | | not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

| No. | '904 Patent Claim 4 | The Reference |
|---|---|---|
| 4[f] | wherein in normal operation, downstream data packets received from the network are passed from the first master unit to each of the daisy chains via the first slave unit in each chain, and upstream data packets received by the slaves in each chain from the subscriber lines are passed via the first slave unit in the chain to the first master unit for transmission over the network. | The Reference discloses wherein in normal operation, downstream data packets received from the network are passed from the first master unit to each of the daisy chains via the first slave unit in each chain, and upstream data packets received by the slaves in each chain from the subscriber lines are passed via the first slave unit in the chain to the first master unit for transmission over the network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 5 | The Reference |
|---|---|---|
| 5 | Apparatus according to claim 4, and comprising a protection interface, which couples the second master unit to the first master unit, and over which interface data packets are conveyed between the first and second master units in case of a fault. | The Reference discloses apparatus according to claim 4, and comprising a protection interface, which couples the second master unit to the first master unit, and over which interface data packets are conveyed between the first and second master units in case of a fault.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

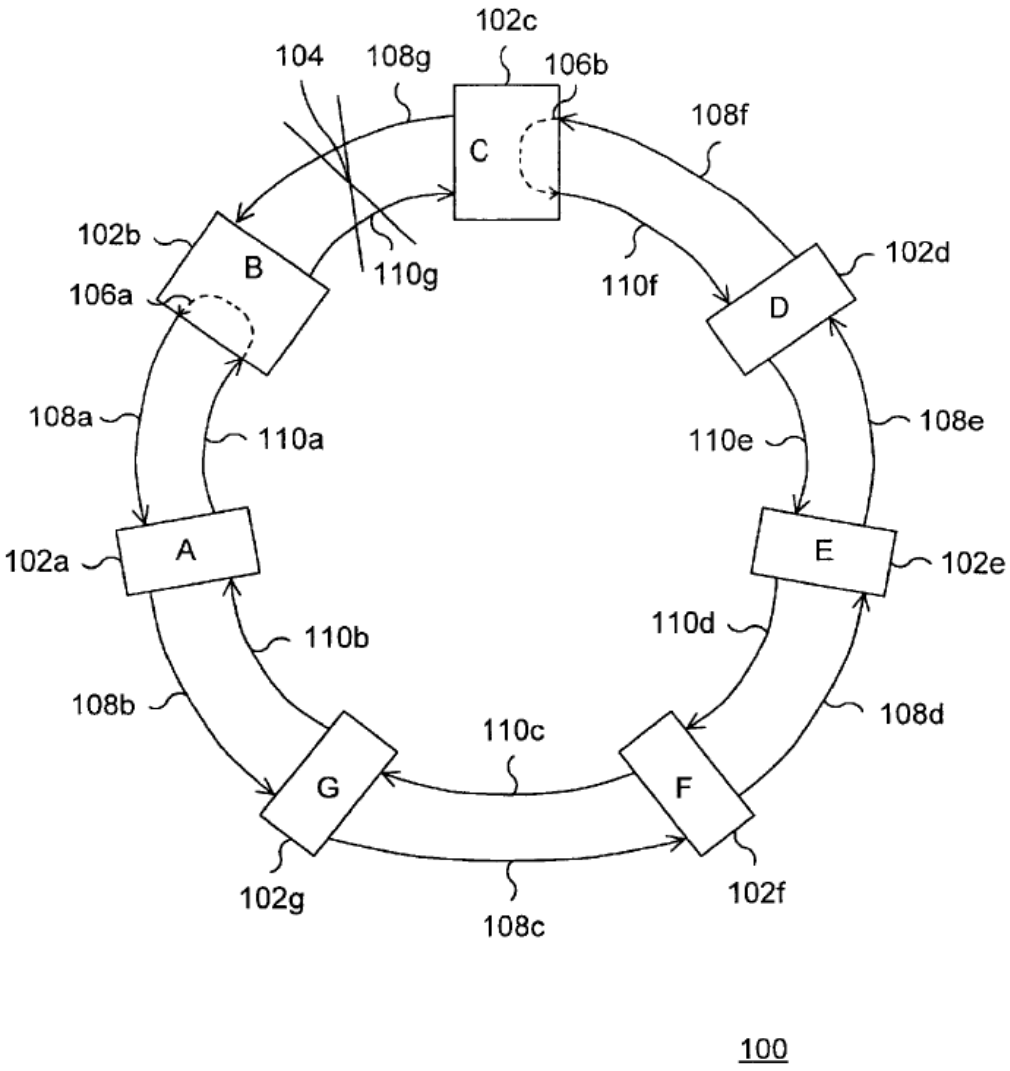
| No. | '904 Patent Claim 6 | The Reference |
|---|---|---|
| 6 | Apparatus according to claim 5, wherein the first master unit bicasts the upstream data packets that it receives from the slave units to the network and, via the protection interface, to the second master unit, which transmits the upstream data packets to the network. | The Reference discloses apparatus according to claim 5, wherein the first master unit bicasts the upstream data packets that it receives from the slave units to the network and, via the protection interface, to the second master unit, which transmits the upstream data packets to the network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

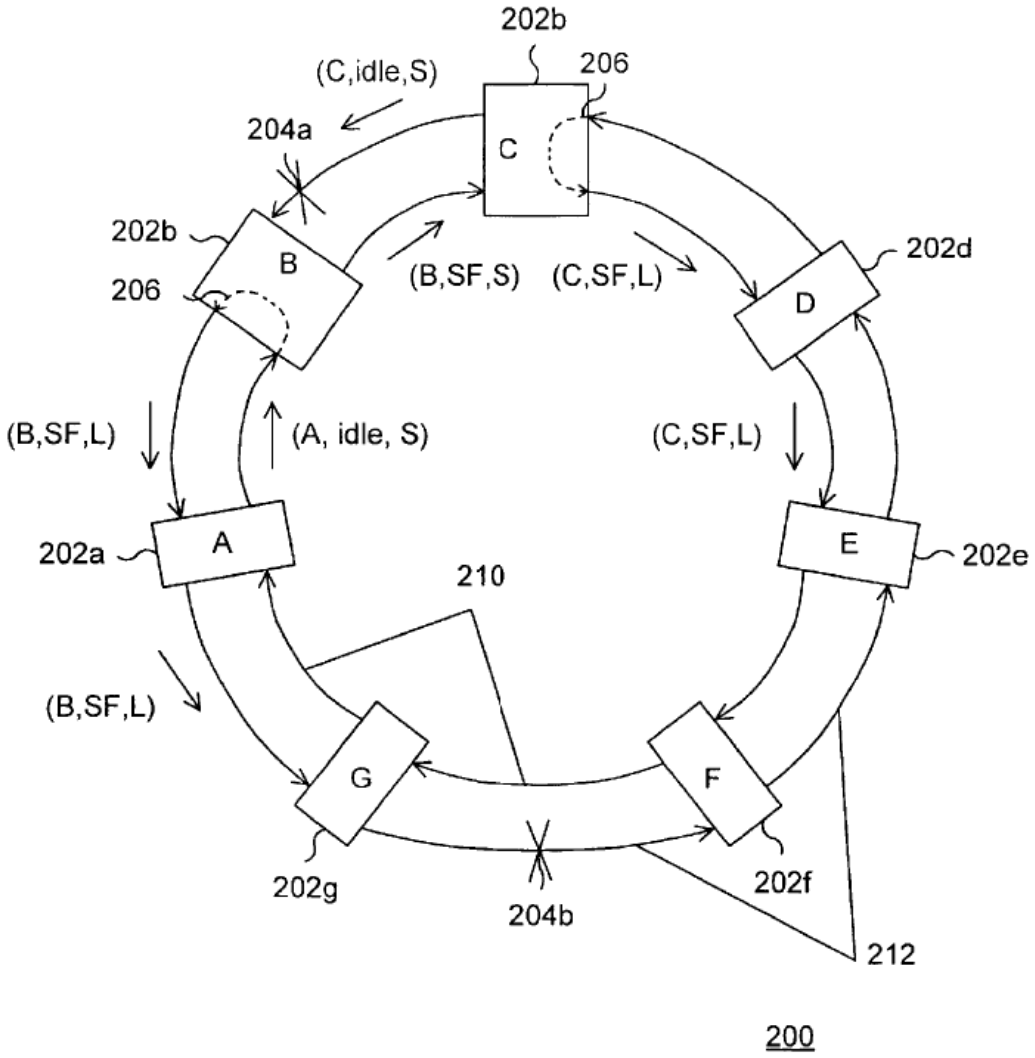
| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| 7 | Apparatus according to claim 4, wherein in case of a fault at a location in one of the daisy chains, data flow in a portion of the daisy chain between the location of the fault and the second master unit is reversed, so that the downstream data packets are passed from the second master unit to the slave units in the portion of the daisy chain via the last slave unit in the chain, and the upstream data packets are passed by the last slave unit to the second master unit. | The Reference discloses apparatus according to claim 4, wherein in case of a fault at a location in one of the daisy chains, data flow in a portion of the daisy chain between the location of the fault and the second master unit is reversed, so that the downstream data packets are passed from the second master unit to the slave units in the portion of the daisy chain via the last slave unit in the chain, and the upstream data packets are passed by the last slave unit to the second master unit.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

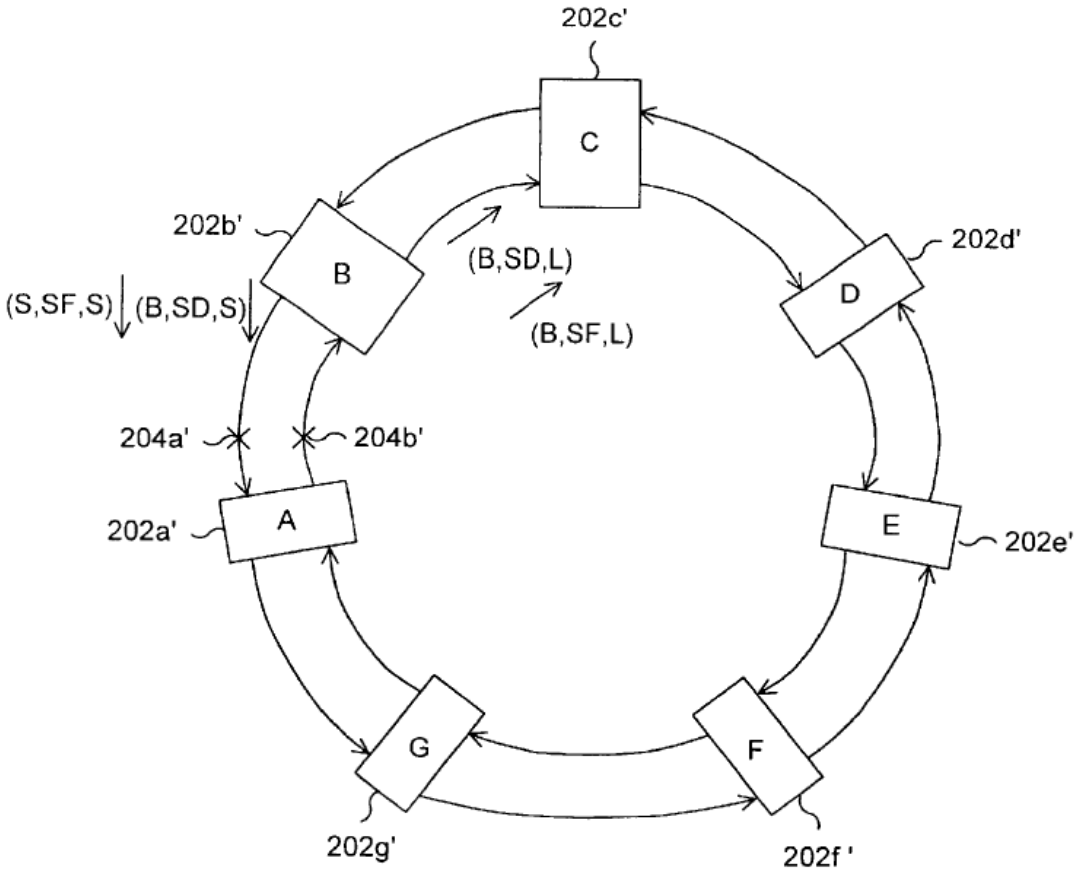
| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotated).<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line Is interrupted."). <br><br> Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") <br><br> Cisco commercialized and patented technology relating to monitoring, detecting, and resolving faults without requiring a network reconfiguration before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include: <br>• Daruwalla <br>• Nederveen <br>• Slater '421 <br>• Petersen |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | **Daruwalla discloses**:<br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. The present invention provides a protection protocol which simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol will appropriately remove a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, would remove protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, Abstract. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 1. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 2. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 11.<br><br>"The present invention relates to computer networks. In particular, the present invention relates to a system and method for providing a protection protocol for fault recovery for a two line bi-directional ring network." Daruwalla, 1:8-11. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 1 shows an example of a two line bi-directional ring network. The ring network 100 is shown to include nodes 102 a-102 g. Each node is typically a computer with embedded processors and at least one network connection. Each node 102 a-102 g is shown to be bidirectionally coupled to two neighboring nodes 102 a-102 g via an inner connection ring 110 a-110 g and an outer connection ring 108 a-108 g. For instance, node 102 a is bidirectionally coupled to nodes 102 b and 102 g. The example of FIG. 1 also shows a problem 104 in the connection between node 102 b and node 102 c. When a problem is detected (such as a bi-directional line cut), the connection between nodes 102 b and 102 d wraps back upon itself, as shown by wraps 106 a and 106 b. In this manner, the connection problem 104 can be avoided." Daruwalla, 1:17-30.<br><br>"In a conventional SONET network, each message sent by a sending node to a receiving node typically needs the identification and location of the receiving node to arrive at the proper destination. Accordingly, manual configuration is typically needed in each node to store the identity and location of each other node in the ring network in order to provide for communication between the nodes in the network." Daruwalla, 1:31-44.<br><br>"In summary, for the protection mechanism to operate, each node needs to know the current ring map (current ring topology). What is needed is a system and method for providing fault recovery for two line bi-directional ring network that minimizes the need to keep track of other nodes in the ring network. Preferably, the system would not require reconfiguration of an internal map of the network when a new node is added to, or existing nodes are removed from the network. The present invention addresses such a need." Daruwalla, 2:23-31.<br><br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. According to the embodiment, when the problem is identified, the node sends a message identifying the problem to a second neighbor which is located at least one node away from the problem. The second neighbor then forwards the message to a third neighbor, unless the second neighbor is dealing with a situation that is higher in a hierarchy of situations than the problem described |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | in the message by the original node. In general, if the second neighbor's situation has a higher priority than the situation described by the original node, then the message is ignored and not forwarded. If, however, the message sent by the original node describes a situation with a higher priority than the situation being dealt with by the second neighbor, then, in general, the second neighbor's situation is ignored, at least for the moment, and the original node's message is forwarded to the next neighbor. In general, a higher priority request preempts a lower priority request within the ring. Exceptions are noted as rules of the protection protocol." Daruwalla, 2:35-58.<br><br>"The present invention provides a protection protocol that simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol automatically appropriately removes a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, removes protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, 2:59-3:3.<br><br>"A method according to an embodiment of the present invention for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:4-12.<br><br>"In another aspect of an embodiment of the present invention, a method for adding a new node to a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node prior to an addition of the new node; initiating a fault recovery procedure when the second node receives the first message; receiving a second |

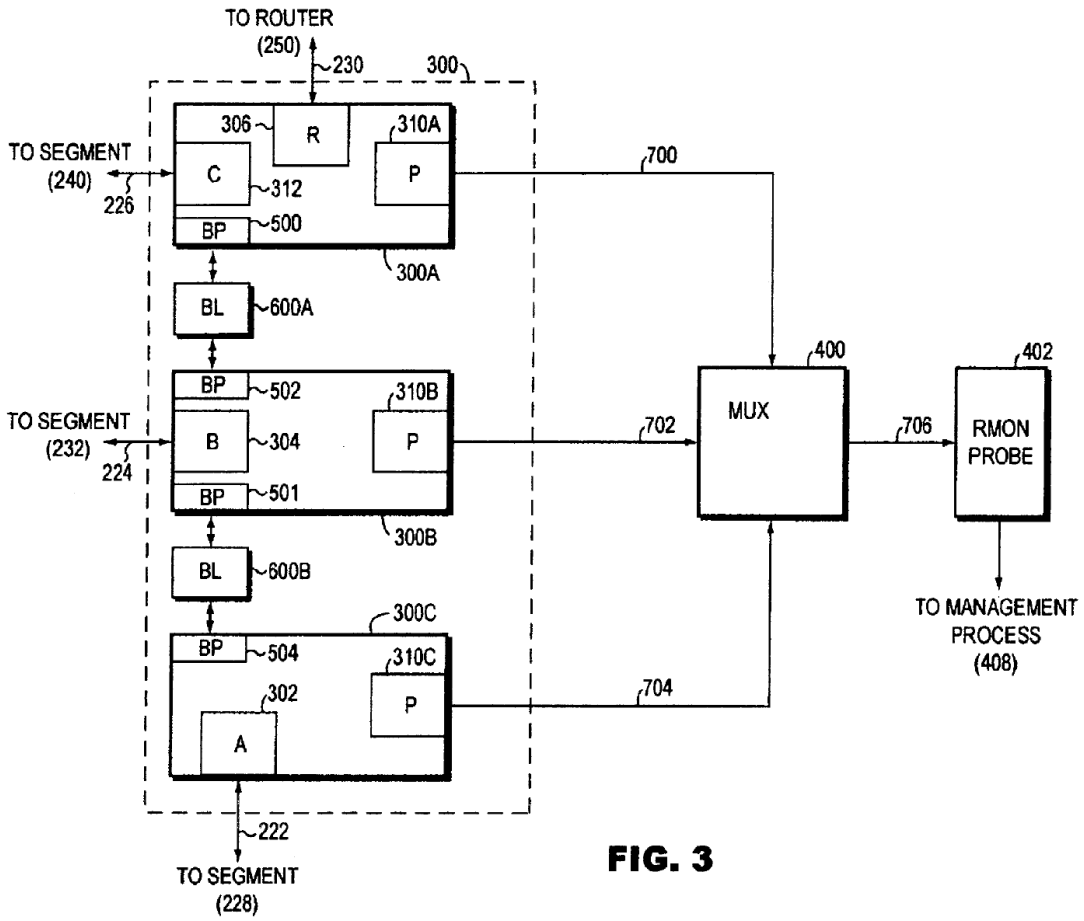
| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | message from the new node; and entering an idle state when the second message is received." Daruwalla, 3:13-24.<br><br>"In yet another aspect of an embodiment of the present invention, a system for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The system comprises means for detecting a situation by a first node, wherein the first node is one of the plurality of nodes; means for sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and means for initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:25-35.<br><br>"FIG. 2 is block diagram of a ring network utilizing a protection protocol according to an embodiment of the present invention." Daruwalla, 3:40-42.<br><br>"FIGS. 4-6 are flow diagrams illustrating various rules within the protection protocol according to an embodiment of the present invention." Daruwalla, 3:46-48.<br><br>"FIGS. 8-12 are flow diagrams and a system diagram illustrating further rules of the protection protocol according to an embodiment of the present invention." Daruwalla, 3:52-54.<br><br>"FIG. 2 is a block diagram showing a ring network system utilizing a method of fault recovery according to an embodiment of the present invention. The ring network 200 is shown to include nodes 202 a-202 g. The nodes 202 a-202 g are shown to be coupled via an inner ring 210 in which the data flows in one direction, such as a clockwise direction. Additionally, the nodes 202 a-202 g are also shown to be coupled by an outer ring 212 in which data can flow in the opposite direction to the inner ring 210, such as in a counter-clockwise direction. The ring network 200 is shown to have a situation 204 a that requires protection, such as a ring wrap 206." Daruwalla, 5:35-45. |

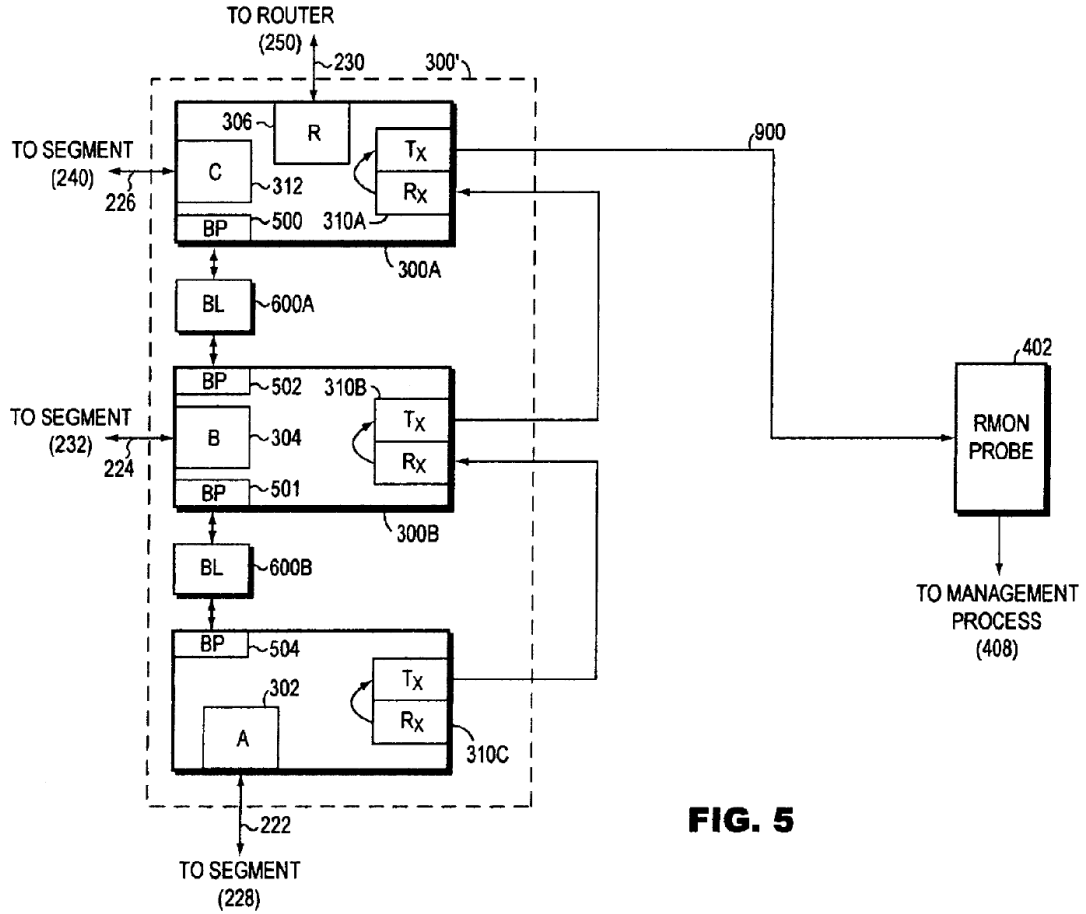
| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 4 is a flow diagram of an example of a method according to an embodiment of the present invention implied by Rules 1-22. An APS packet is received via step 400. It is determined whether the APS packet has been sent along a long path via step 402. If the packet was not sent via a long path, then the APS packet is not forwarded via step 406. Accordingly, if the APS packet was sent via the short path, then the packet is not forwarded via step 406. If, however, the packet was sent through the long path via step 402, then the APS packet may be forwarded via step 404. Note that for this example of Rule (1), it is assumed that the long path message does not have to pass through a wrapped connection in order to be forwarded. Otherwise, if the long path message needs to pass through a wrapped connection in order to be forwarded, then the message will simply not be forwarded." Daruwalla, 6:21-36.<br><br>"FIG. 6 is a flow diagram illustrating Rules 4 and 5. A node detects a problem between the node and a first neighbor via step 600. The node performs a wrap away from the side on which the problem exists via step 602. A short path message is then sent to the first neighbor informing it of the problem via step 604. Additionally, a long path message is also sent to a second neighbor informing the second neighbor of the problem via step 604. The first neighbor then performs a wrap away from the side of the problem via step 606. The first neighbor also sends an IDLE message, indicating a wrapped status, on a short path to the node that detected the problem via step 608. This message is sent across the failed span. Note that IDLE messages do not get wrapped and are sent across failed spans if possible. Additionally, the first neighbor also sends a message on a long path toward the side without the problem via step 608." Daruwalla, 6:64-11.<br><br>"An example of the method described in FIG. 6 can be seen in FIG. 2. Node 202 b has detected a problem 204 a and performs a wrap 206 on the side on which the problem exists. Node 202 b also sends a short path message to the neighbor on the other side of the problem 204 a, which is node 202 c. Node 202 b also sends a long path message to its other neighbor node 202 a informing it of the problem. Node 202 c performs a wrap 206 on the side of the problem and sends an IDLE message on a short path to node 202 b. Node 202 c also sent a message on a long path toward the side without the problem to its neighbor 202 d." Daruwalla, 7:12-21. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 7 lists the hierarchy of priorities of Rule (8). For ease of reference, the hierarchy is separated into Class I-III. Class I is the highest priority, while Class III is the lowest priority. An example of a highest priority message in Class I is lockout. Lockout is an order stating that the ring network is not to wrap under any circumstances." Daruwalla, 7:22-26.<br><br>"Examples of the next priority listed in Class II are forced switch and signal fail. Forced switch indicates that the ring network is configured to wrap at the point of the forced switch. Signal fail is a situation where either two nodes cannot communicate with each other, or one node cannot hear the other node. An example of a signal fail is a physical break in the communication lines between two nodes." Daruwalla, 7:27-33.<br><br>"Note that members of Class II can co-exist (Rule 9). For example, multiple forced switches and signal fails can co-exist. Additionally, members of Class I can co-exist (Rule 10). For example, multiple lockouts in a single ring network can co-exist. However, situations in Class III cannot co-exist with other situations (Rule 11). For example, a signal degrade cannot co-exist with a wait-to-restore." Daruwalla, 7:52-58.<br><br>"When there are multiple requests of the same priority within Class III, the first request to complete a long path signaling will take priority (Rule 13). For example, if there are two signal degrades located on the same ring network, then the first signal degrade which completes the long path signaling will take priority over the other signal degrade. By not allowing members of Class III to co-exist with one another, partitioning of the ring network is avoided." Daruwalla, 7:59-65.<br><br>"In case of two equal requests within Class III on both inner and outer rings of the ring network, the tie is broken by choosing a request on one of the rings, such as the outer ring request (Rule 14). For example, if a signal degrade occurs both on the inner and outer rings, then a request on a predetermined ring, such as the outer ring, will take priority over the other requests." Daruwalla, 7:66-8:5. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 8 is a flow diagram illustrating Rules (9), (10), (11), (13), and (15). Note that the flow diagram described in FIG. 8 is merely an example of one way in which the rules of the method according to the embodiment of the present invention can be executed. For example, the determination of whether the long path message is a Class I request via step 802 or a Class II request via step 810 can be in reverse order." Daruwalla, 8:6-11.<br><br>"A wrapped node receives a long path message via step 800. It is then determined if the long path message is a Class I request via step 802. The classes used in FIG. 8 are meant to correspond with the example of classes defined in FIG. 7. If the long path message is a Class I request, then it is determined if a local situation also has a Class I request via step 804. A local situation includes scenarios such as when a node detects a situation or problem, or when a node is made aware of a problem or situation via a short path message from its neighbor. If a local situation is not a Class I request via step 804, then any existing local wraps are unwrapped and the long path message is forwarded via step 806. If, however, a local situation is a Class I request via step 804, then the connections are already unwrapped or was never wrapped, and the long path message is forwarded via step 808." Daruwalla, 8:12-26.<br><br>"FIG. 12 is a flow diagram illustrating rules (20) and (21) of the method according to the embodiment of the present invention. A wrapped node determines that a problem has been cleared via step 1200. It then enters a wait-to-restore state via step 1202. It is then determined if its neighbor is the same neighbor as previously noted via step 1204. The node can save the source of a short path message at the time of wrap initiation to note the identity of its neighbor. If the current neighbor is not the same as the previous neighbor via step 1204, then an IDLE state is entered via step 1206. If, however, the current neighbor is the same as the previous neighbor via step 1204, then it is determined whether a pre-determined wait-to-restore time has expired via step 1208. Once the pre-determined wait-to-restore time has expired, then the node enters an IDLE state via step 1206." Daruwalla, 12:60-13:6.<br><br>"A method and system for fault recovery for a two line bi-directional network has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor." Daruwalla, 13:7-19. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | **Nederveen discloses**:<br>"A technique for use in gathering network activity-related information from cascaded network switches is provided. Using this technique, the information can be gathered without substantially reducing performance of the cascaded switches. In one embodiment, a single remote monitoring probe is connected via respective connections to each of the switches so as to receive the information from the switches. In another embodiment, only one of the switches is connected to the probe, and the other switches transmit their respective portions of the information to the switch connected to probe. The switch connected to the probe provides these portions of the information, as well as, any of its respective activity-related information to the probe. In this latter embodiment, the switches may be connected by dedicated connections and switch ports that are used solely for communicating the activity-related information." Nederveen, Abstract. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>**FIG. 3**<br>Nederveen, FIG. 3. |

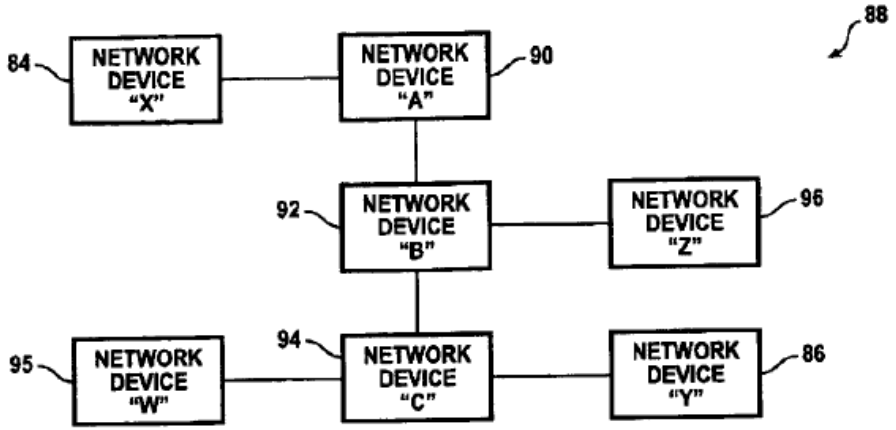
| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>**FIG. 5**<br><br>Nederveen, FIG. 5.<br><br>"Thus, it would be desirable to provide a stacked switch monitoring technique that permits efficient offloading of raw data processing from the stacked switches, requires only a minimal number of specialized network entities to gather and process such raw data, and does not result in substantial degradation of stacked switch performance." Nederveen, 4:38-43. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Accordingly, the present invention provides a technique for remote monitoring of a switch network that overcomes the aforesaid and other disadvantages and drawbacks of the prior art. More specifically, in one aspect of the present invention, a technique is provided for gathering information that may be useful in network management (e.g., switch port activity-related information), from switches in the network that are in a stacked configuration. The information is gathered from the stacked switches by a single network entity (e.g., an SNMP remote monitoring probe) in such a way that it does not substantially degrade the performance of the switches. This is accomplished, in one embodiment of the technique of the present invention, by connecting the switches via respective connections to a multiplexer that selectively connects the switches, according to an arbitration scheme, to the single network entity. The entity gathers respective portions of the information from switches when it is connected to the switches by the multiplexer. The information gathered by the entity may be provided to another network entity (e.g., an SNMP management node) in order to permit the other entity to use that information in managing the network." Nederveen, 4:46-67.<br><br>"In another embodiment of the technique of the present invention, only one of the switches is connected to the single information gathering entity. The switches that are not connected to the entity transmit, via respective dedicated ports and connections (i.e., ports and connections that are used solely for network information gathering activities), their respective portions of the information to the switch that is connected to the entity. The switch that is connected to the entity transmits, via a respective dedicated port and connection, the information received from the other switches, as well as, its own information to the entity." Nederveen, 5:1-11.<br><br>"FIG. 3 is a schematic, functional block diagram illustrating in greater detail the construction of the stacked switch network shown in FIG. 2." Nederveen, 5:26-28.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network configured to employ another embodiment of the present invention." Nederveen, 5:32-34. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIGS. 2-5 illustrate features of a computer network 200 wherein embodiments of the present invention may be advantageously practiced. Network 200 comprises a stacked switch network 300 which interconnects a plurality of network segments 228, 232, 240, and 251. Each segment 228, 232, 240 comprises one or more local area networks having computer endstations (not shown). Segment 251 is a network router segment that comprises network router 250. Each segment 228, 232, 240 is coupled via a respective communications link 222, 224, 226 to a respective port 302 (i.e., port A), 304 (i.e., port B), 312 (i.e., port C) of the switch network 300. Likewise, the router 250 of router segment 251 is coupled via a respective trunk line 230 to router port 306 (i.e., port R)." Nederveen, 5:46-59.<br><br>"Stacked switch network 300 comprises a plurality of data network switches 300A, 300B, 300C (e.g., Catalyst 3900™ series switches of the type commercially available from the Assignee of the subject application) coupled together via conventional stack link bus connection logic 600A, 600B. More specifically, logic 600A couples a stack link bus port and associated logic 500 in switch 300A to a stack link bus link port and associated logic 502 in switch 300B. Similarly, logic 600B couples another stack link bus port and associated logic 501 in switch 300B to a stack link bus port and associated logic 504 in switch 300C. It should be understood that although, as is shown in FIG. 3, switches 300A and 300B, and switches 300B and 300C, may be coupled serially together by separate respective logic elements 600A, 600B, each of the switches 300A, 300B, 300C may be coupled together via a single respective stack link bus port in the switch to a single stack link bus connection logic block (not shown, e.g., of the type that is commercially available under the tradename Catalyst Matrix™ from the Assignee of the subject application). Further alternatively, depending upon the particular design and functionality of the ports 500, 501, 502, and 504, and the control and forwarding logic (whose operation will be described more fully below) in the switches 300A, 300B, 300C, the circuitry in logic 600A, 600B may instead be comprised in the ports 500, 501, 502, and 504 and/or control and forwarding logic, and therefore, in this alternative configuration, the logic 600A, 600B in the network 300 may be replaced by simple connection means (e.g., cable connectors)." Nederveen, 6:29-57. |

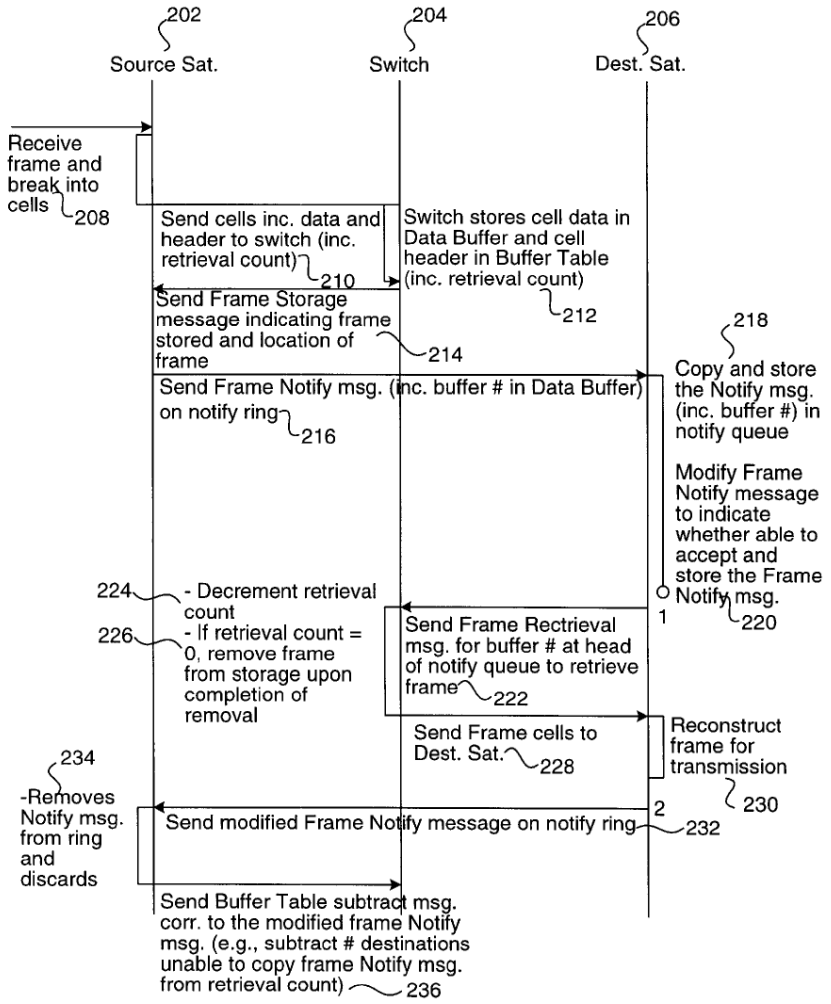
| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Each switch 300A, 300B, 300C includes a respective internal bus (e.g., element 800 in switch 300C) that is coupled via at least one stack link bus port and associated interface logic (e.g., 504 in switch 300C) to external stack link bus connection logic (e.g., element 600B in switch 300C). Each switch 300A, 300B, 300C also includes respective programmable control and forwarding logic (e.g., element 802 in switch 300C) comprising processing, memory, and other circuitry for storing and learning configuration information (e.g., source and destination MAC addresses of messages received by the switch, switch bridging table, switch segments' spanning tree and virtual local area network information, etc.), and for providing appropriate commands to other elements (e.g., the switch ports) to cause data messages received by the switch to be forwarded to appropriate network segments coupled to the switch based upon this configuration information. In each switch, the switch's port logic circuitry (e.g., port A logic 302 and port P logic 310C in switch 300C) and control and forwarding logic are coupled to each other via that switch's respective internal bus. The stack link bus port and associated logic in each switch 300A, 300B, 300C may comprise a Catalyst™ stack port line interface card (commercially available from the Assignee of the subject application) inserted into a bus expansion slot (not shown) in the switch. Although not shown in the Figures for purposes of clarity of illustration, each switch 300A, 300B, 300C in network 300 typically will include tens or hundreds of ports coupled to network segments." Nederveen, 6:58-7:19.<br><br>"The control and forwarding logic and stack link bus port and associated logic in each switch, and the logic 600A, 600B, are configured to together implement conventional techniques for permitting the switches 300A, 300B, 300C to function together as a single logical/virtual switch. More specifically, when configured in the stacked arrangement 300, after the switches 300A, 300B, 300C and logic 600A, 600B are initially activated, they execute initial power-on self-diagnostics, and thereafter, enter a "stack discovery" mode of operation." Nederveen, 7:20-29.<br><br>"In the stack discovery mode of operation, the control and forwarding logic in each switch 300A, 300B, 300C first "senses" that its switch is coupled to logic 600A and/or 600B, and then determines the particular configuration of the stacked switch network 300, using suitable conventional autosensing/autoconfiguration techniques. The control and 96orwarding logic in the switches 300A, 300B, 300C then assigns to the switches respective unique |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | identification numbers (e.g., based upon unique identification numbers of respective ports of the logic 600A, 600B to which the switches are coupled)." Nederveen, 7:30-40.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network 300′ configured to employ another embodiment of the present invention. It should be understood that unless specifically stated to the contrary, the structure and operation of the network 300′ are substantially the same as the structure and operation of network 300. In network 300′, each of the dedicated ports 310A, 310B, 310C comprises a respective transmit portion and receive portion, referenced in FIG. 5 as RX and TX, respectively." Nederveen, 11:20-29.<br><br>**Slater '421 discloses**:<br>"A method and apparatus for discovering paths to other network devices includes a protocol and network management application that can be executed on network devices. The Ethernet protocol is used to detects paths to other network devices, knowing only the Ethernet address of the destination. A discovery protocol is extended to add hop probe and hop probe reply Type-Length-Value fields in a variable-length list. The hop probe fields contain a hop count, a destination Ethernet address, and a source Ethernet address. When a hop probe is received by a network device, the hop count field is decremented by one and the hop probe is forwarded. Packet received with a hop count of one are not forwarded and a hop probe reply is sent back to the Ethernet source address of the hop probe. The hop probe reply fields contain a destination Etherned address and a source Ethernet address." Slater '421, Abstract. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br><br>Slater '421, FIG. 6.<br><br>"Partly as a result of the increased complexity of networks, network administrators must often troubleshoot problems with their network. Two classes of network problems often faced by network administrators are "reachability" problems and performance slowdowns. Reachability problems occur when one or more network devices cannot be accessed through a network, and can be caused by hardware or software failures, cabling problems, or any of several other types of difficult-to-diagnose problems that can occur in a network." Slater '421, 7:11-20.<br><br>"Some of the world's largest networks today rely on the TCP/IP suite of networking protocols. With a relatively small kit of basic tools, network administrators can learn much about an internetwork. "Ping" and "traceroute" commands, "show" commands, and "debug" commands (all of which are typically available via the basic management interface on a network device) form the core of the network administrator's internetwork toolkit. Ping and traceroute commands can be useful tools in determining where failures are occurring, but they are cumbersome to use, and require knowledge of the IP address or host name of the |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | destination network device. The show commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data. Finally, debug commands provide clues about the status of a network device and other network devices directly or indirectly connected to it. Because debug commands can create performance slowdowns, they must be used with great care, and using the wrong debug command at the wrong time can exacerbate problems in already poorly performing networks." Slater '421, 7:55-8:8.<br><br>"Embodiments of the present invention as illustrated herein use the Cisco™ Discovery Protocol ("CDP") to automatically detect paths to specified network devices in Ethernet LANs. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task." Slater '421, 9:10-15.<br><br>"CDP is a media-independent device discovery protocol which can be used by a network administrator to view information about other network devices directly attached to a particular network device. In addition, network management applications can retrieve the device type and SNMP-agent address of neighboring network devices. This enables applications to send SNMP queries to neighboring devices. CDP thus allows network management applications to discover devices that are neighbors of already known devices, such as neighbors running lower-layer, transparent protocols." Slater '421, 9:16-26.<br><br>"It is to be understood that the present invention is not limited to devices that are compatible with CDP. CDP runs on all media that support the Subnetwork Access Protocol ("SNAP"), including LAN and Frame Relay. CDP runs over the data link layer only. Each network device sends periodic messages to a multicast address and listens to the periodic messages sent by others in order to learn about neighboring devices and determine when their interfaces to the media go up or down. Each device also advertises at least one address at which it can receive SNMP messages. CDP messages, or "advertisements," contain holdtime information, which indicates the period of time a receiving device should hold CDP information from a neighbor before discarding it. With CDP, network management applications can learn the device type and the SNMP-agent address of neighboring devices. This process enables applications to send SNMP queries to neighboring devices." Slater '421, 9:27-43. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "It should be noted that-normally, CDP packets according to aspects of the present invention are transmitted at regular intervals (e.g. once every 60 seconds). However, in embodiments of the present invention, when a Hop Probe or Hop Probe Reply needs to be forwarded by a network device, the network device is commanded to send a CDP packet immediately." Slater '421, 16:66-17:5.<br><br>"The present invention is much faster than the previous method that involved logging in to each intermediate network device, entering the "show cdp neighbors" command, and interpreting the output to find the next hop along the path to the destination network device. Also, the present invention allows individual users, such as network administrators, to execute a tool to manually discover paths through a network of Ethernet switches. The present invention can be used by network management software to automatically map the topology of clusters of network devices, such as Ethernet switches. Finally, the present invention is useful in loop detection. Enhancements to Spanning Tree and other bridge-level routing protocols can test proposed changes to switch topology prior to making them." Slater '421, 17:6-20.<br><br>**Petersen discloses**:<br>"Methods and apparatus for enabling communication between a source network device and one or more destination network devices are disclosed. A system enabling communication between a source network device and one or more destination network devices includes a switch and a ring interconnect. The switch is adapted for connecting to the source network device and the one or more destination network devices. More particularly, the switch is capable of storing data provided by the source network device and retrieving the data for the one or more destination network devices. The ring interconnect is adapted for connecting the source network device and the one or more destination network devices to one another. In addition, the ring interconnect is capable of passing one or more free slot symbols along the ring interconnect. Thus, the ring interconnect is capable of expanding when one or more of the free slot symbols are each replaced by a frame notify message indicating that the data has been stored by the switch for retrieval by the one or more destination network devices." Petersen, Abstract. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>***FIG. 2***<br>Petersen, FIG. 2. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "The present invention relates to a mixed topology data switching system that combines a radial interconnect with a ring interconnect. More particularly, the radial interconnect permits devices to store and retrieve data using a switch, while the ring interconnect permits devices along the ring interconnect to provide notification that data has been stored for retrieval, as well as provide feedback regarding the ability or inability to retrieve such data." Petersen, 1:34-41.<br><br>"In controlling the flow of network traffic through a switching system, it is often desirable to provide feedback to the source of the data. For instance, although a transmitting device, hereinafter referred to as a "source device," may transmit or forward data to a receiving device, hereinafter referred to as a "destination device," the destination device may be incapable of handling the data. In these circumstances, the source device is often unaware that the data was not accepted by the destination device, complicating switch management. Common solutions to the problem of switch traffic management have included ensuring that all intended destination devices are "ready to receive" prior to transmitting data on a ring or bus interconnect, or insisting that each intended destination device send an explicit acknowledgement back to the source device. Both of these approaches result in reduced efficiency of the interconnect scheme. By way of example, in a ring network, such acknowledgment is typically provided in the data frame being transmitted. As another example, in other interconnect schemes, each such device may send a separate acknowledgment, therefore adding to the traffic on the network. Accordingly, it would be desirable if a traffic management scheme were established which could provide such feedback to the source of the data while minimizing traffic management overhead." Petersen, 2:8-32.<br><br>"According to one embodiment, the present invention combines the use of two data transport methods: a point-to-point radial interconnect and a ring interconnect. The radial interconnect connects interface devices to each other through the services of a central switch device to permit the transport of data. Typically, a single interface has a single dedicated radial interconnect to the central switch. These interface devices are further connected to one another via a ring interconnect to convey retrieval notifications regarding forwarding of the data (by source devices) and receipt of the data (by destination devices)." Petersen, 2:36-46. |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Each radial interconnect provides a narrow, high baud-rate connection to convey to the actual data from and to the associated interface without being burdened by the unrelated traffic for the remaining interfaces in the system. This is accomplished through the use of a central switch device, which stores and retrieves data for the various interfaces in the system. As will be apparent from the following description, this architecture provides numerous advantages over a wide parallel bus or ring." Petersen, 2:47-55.<br><br>"The ring interconnect may be used to convey a "retrieval notification"(i.e., retrieval message) that may be observed by all potential retrieving interfaces. The retrieval notification notifies specific devices ("destination devices") or interfaces that one or more frames addressed to them are available from the switch device. Moreover, the ring interconnect permits each destination device to provide feedback to the source device letting the source know whether the destination has accepted the notification provided by the source device and therefore whether the destination can retrieve the data intended for it. The feedback is particularly useful in buffer management applications. In this manner, an efficient and flexible data transport and retrieval notification system that includes a feedback path to the source of the data is provided." Petersen, 2:56-3:3.<br><br>"FIG. 2 is a process flow diagram illustrating a method of providing network communication according to an embodiment of the invention." Petersen, 3:9-11.<br><br>"FIG. 2 is a process flow diagram illustrating in further detail a method of providing network communication in the above-described system according to an embodiment of the invention. As shown, process steps performed by a source device 202 are illustrated along an associated vertical line, steps performed by a switch 204 are illustrated along another vertical line, and steps performed by a destination device 206 are illustrated along still another vertical line." Petersen, 4:50-57.<br><br>"When the frame is stored by the switch 418, the source device preferably receives an acknowledgment that the data has been stored. Thus, to provide this feedback, a frame storage message (i.e., storage reply) is sent from the switch 418 on the channel 416 to the channel interface 414. The frame storage message is then provided to the notify ring interface as |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | shown at 430 and sent on the notify ring. Once this acknowledgment is received by the interface device 402, the designated destination devices may be notified via notify ring interface 424. As described above, a Frame Notify message may be sent via the notify ring interface 424 to the destination devices. More particularly, the Frame Notify message may identify one or more destination devices for the frame and specify the location of the frame to be retrieved. By way of example, the location of the frame to be retrieved by the destination devices may be designated by a buffer number 430. In addition, the destination devices for the frame may be specified in the Frame Notify message through a notify queue map 426. More particularly, the notify queue map 426 may specify a notify queue associated with a particular destination device. The notify queue may be expressly designated through the use of one or more bits as well as implied through the specification of a priority level for the data. The notify queue map 426 will be described in further detail with reference to FIG. 13. The notify ring interface 424 then creates a Frame Notify message including the notify queue map 426 and the buffer number 430 which is then sent on an outbound interface of the notify ring 432." Petersen, 7:55-8:16.<br><br>"As described above, the notify ring may be expanded to accommodate communication between interface devices. The communication between the interface devices and the switch is therefore performed on one or more channels rather than the notify ring. As a result, the flexibility of the notify ring does not effect the speed with which the interface devices may communicate with the switch. Thus, where a single port operates at a faster speed than the channels, multiple channels may be grouped together. In this manner, the speed with which the switch may communicate with the interface devices may be maximized." Petersen, 20:27-36.<br><br>"The present invention provides a mixed topology data switching system that combines a point-to-point radial interconnect with a ring interconnect to maximize the speed of network traffic. The radial interconnect provides a narrow, high baud-rate connection to convey the data traffic for just the interface in question, without being burdened by all of the unrelated traffic for the remaining interfaces in the system. At the same time, the ring interconnect permits retrieval notifications to be observed by all potential retrieving interfaces. The ring topology further permits each destination interface to provide feedback to the source |

| No. | '904 Patent Claim 7 | The Reference |
|---|---|---|
| | | interface, which is valuable for buffer management applications. Moreover, the point-to-point ring topology bus employs a variable latency access method that enables messages to be passed across the bus with low latency when the system is quiet and with increased latency when the system is busy. In addition, since control messaging around the ring interconnect and across the channel interconnects are embedded in the data stream, the number of pins required and manufacturing costs are reduced." Petersen, 20:38-57. |

| No. | '904 Patent Claim 8 | The Reference |
|---|---|---|
| 8 | Apparatus according to claim 7, wherein the downstream packets for the slave units in the portion of the daisy chain between the location of the fault and the second master unit are conveyed to the second master unit from the first master unit via another one of the daisy chains. | The Reference discloses apparatus according to claim 7, wherein the downstream packets for the slave units in the portion of the daisy chain between the location of the fault and the second master unit are conveyed to the second master unit from the first master unit via another one of the daisy chains.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling and Sugawara.<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 1:26-45 ("FIG. 1 illustrates a conventional data link protecting system. A through D designate normal lines and P designates a backup line. LIUA through LIUD and LIUa through LIUd denote line interface circuits having functions of communicating with opposed devices, detecting the error rate of received signals, detecting abnormalities in main signals, such as signal interruption, and feeding intermediate repeaters. MUXA through MUXD and MUXa through MUXd designate multiplexing/demultiplexing circuits, and SW1 and SW2 denote switch circuits adapted to switch an abnormal line to the backup line. MPU1 and MPU2 denote supervisory and control units which control their associated devices, supervise |

| No. | '904 Patent Claim 8 | The Reference |
|---|---|---|
| | | the line interface circuits LIUs to switch the switch circuits SW as needed and perform data link communication with opposed devices. Note that the FIG. 1 illustrates only the arrangement for data transmission in one direction. The arrangement for data transmission in the opposite direction is the same as above.”). |

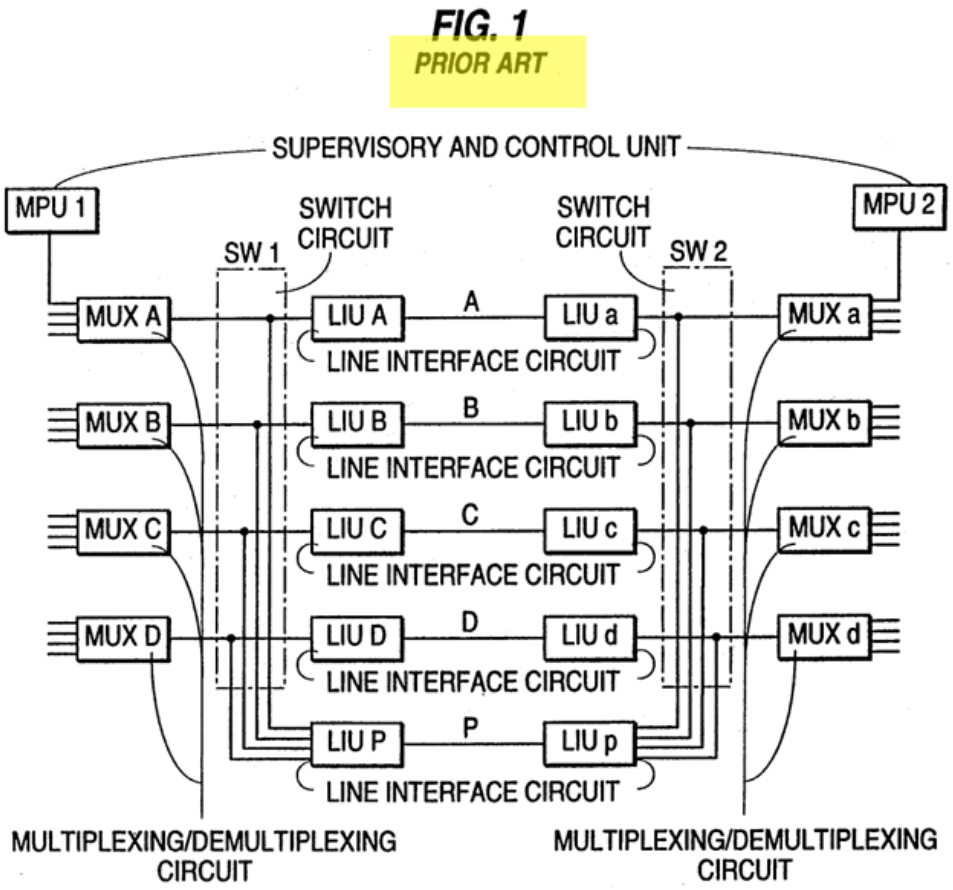



FIG. 1 (annotated).

| No. | '904 Patent Claim 8 | The Reference |
|---|---|---|
| | | Sugawara, 3:24-39 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPU1 not shown. Responsive to this, the supervisory and control unit MPU1 switches switch circuit SW1 to connect MPU1 to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B."). |

FIG. 2

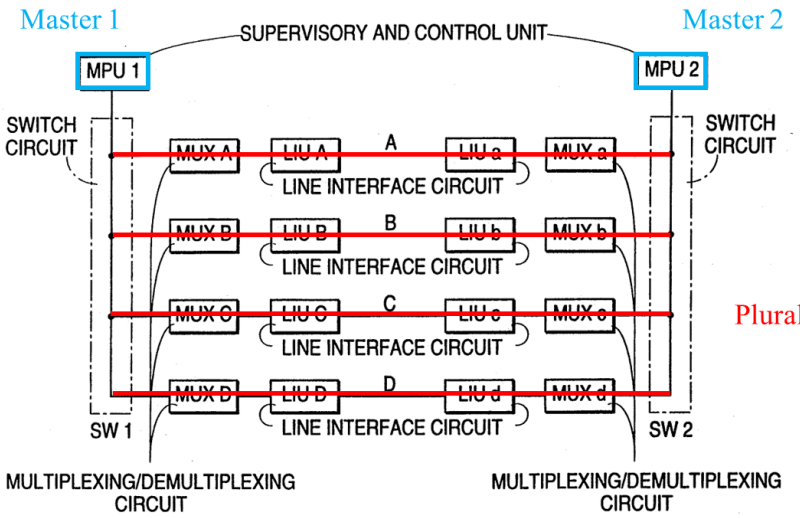



FIG. 2 (annotation added).

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| 9[preamble] | Network access apparatus, comprising: | The Reference discloses network access apparatus, comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[preamble]. |
| 9[a] | first and second master units, each comprising a physical interface to a packet-switched network; | The Reference discloses first and second master units, each comprising a physical interface to a packet-switched network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[a].<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626 |

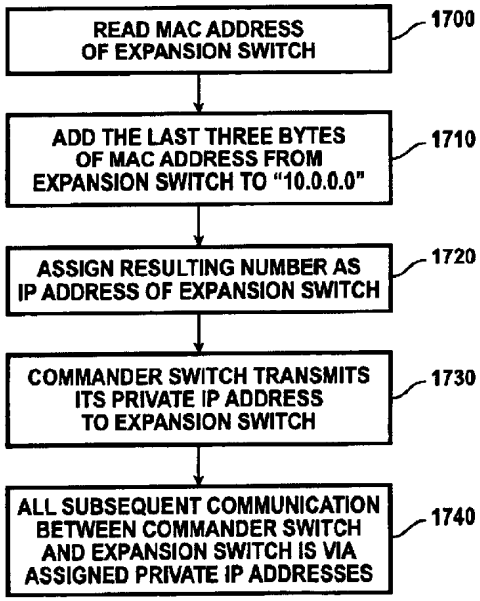
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | • Duuvury '820<br><br>**<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 – Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2. |

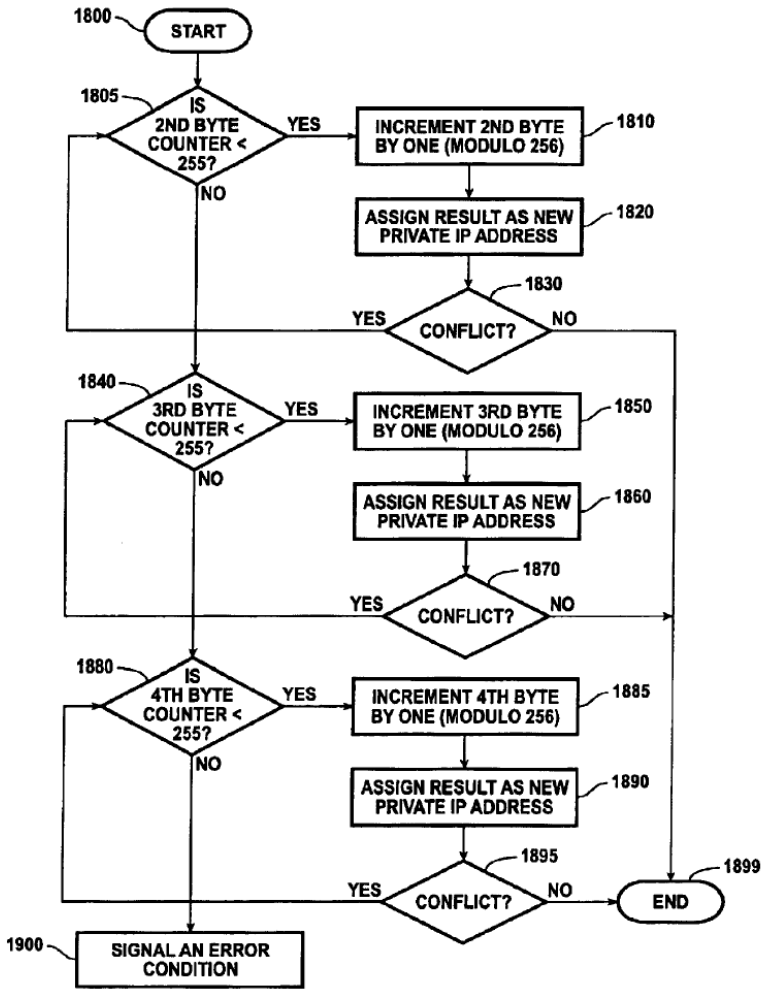
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. Software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

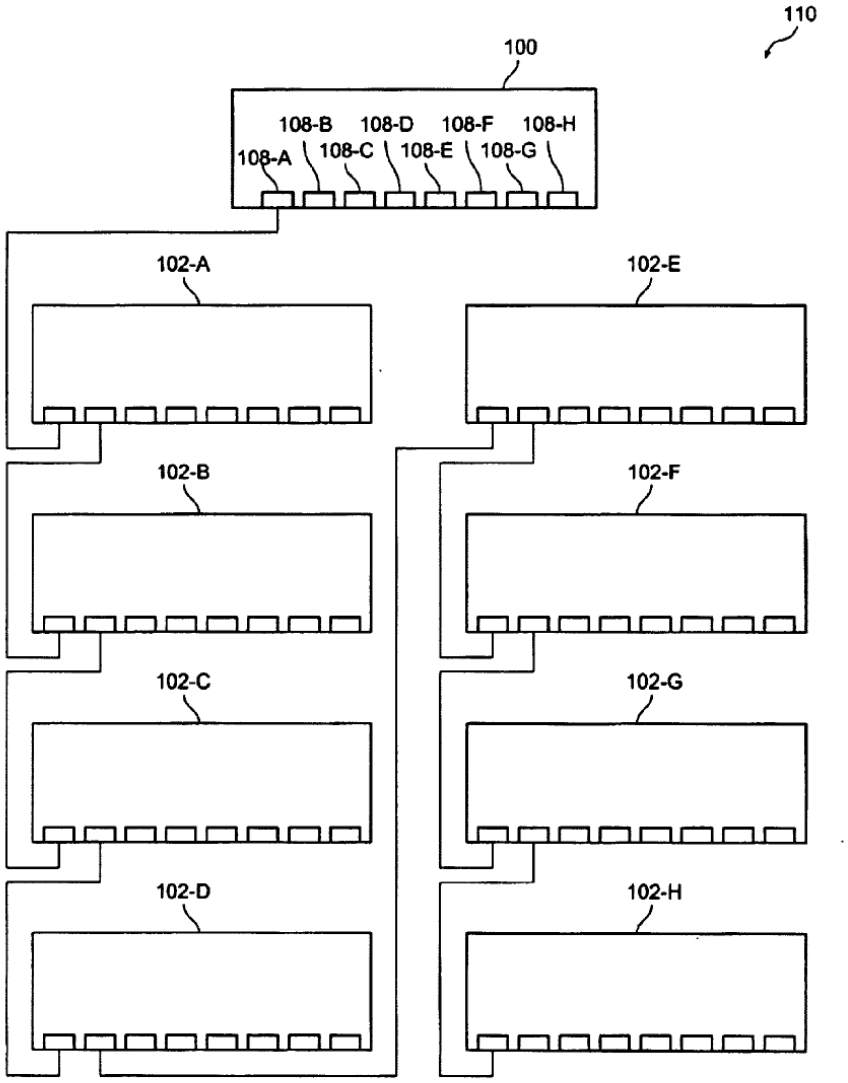
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

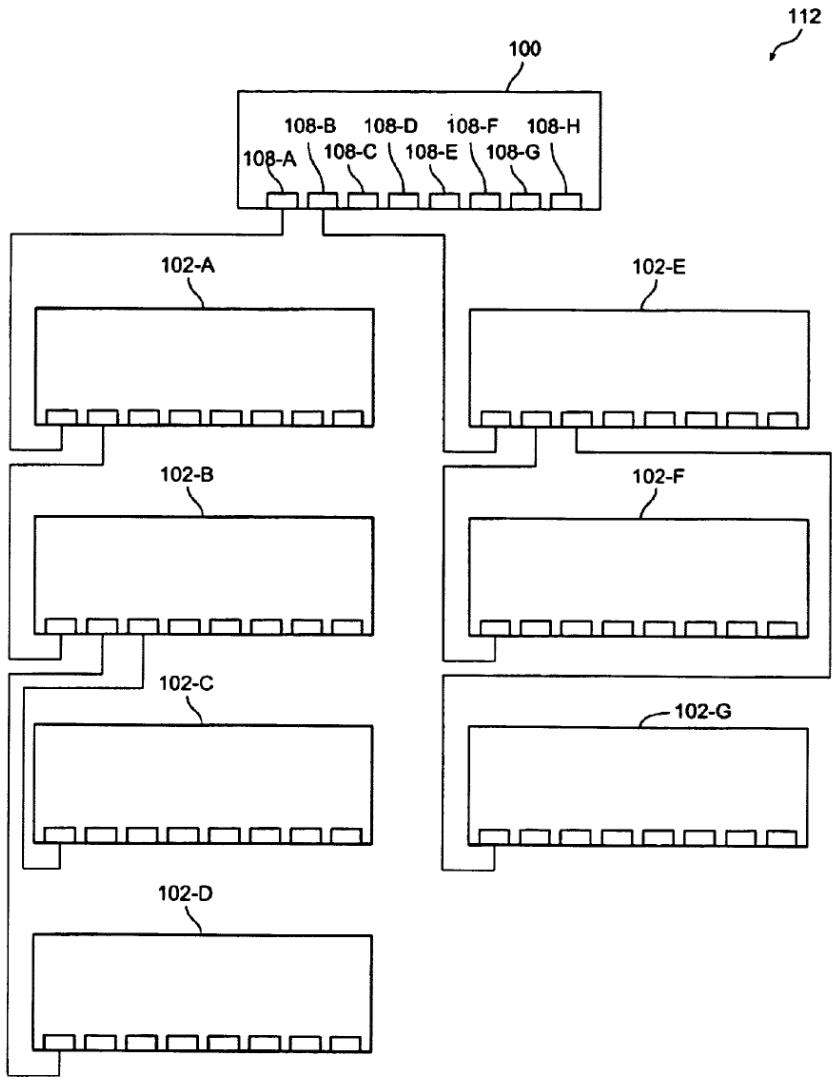
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 9[b] | a plurality of slave units, | The Reference discloses a plurality of slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[b].<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796 |

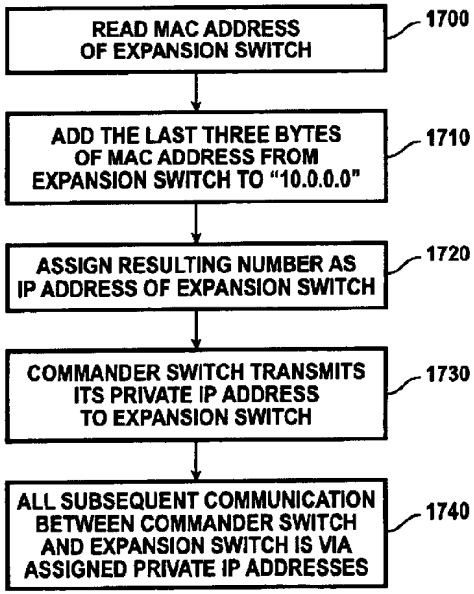
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | • Slater '433<br>• Duvvury '626<br>• Duuvury '820<br><br>Cisco Catalyst Press Release discloses:<br>"May 24, 1999 – Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to |

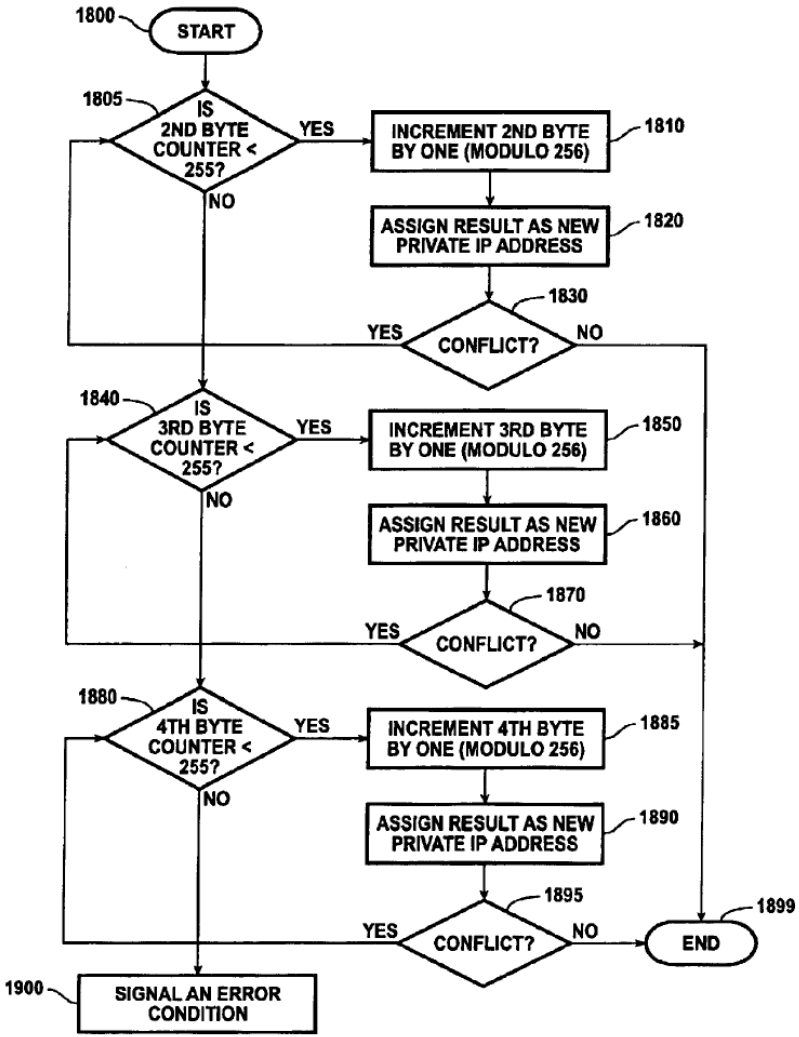
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2.<br><br>"With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. Software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

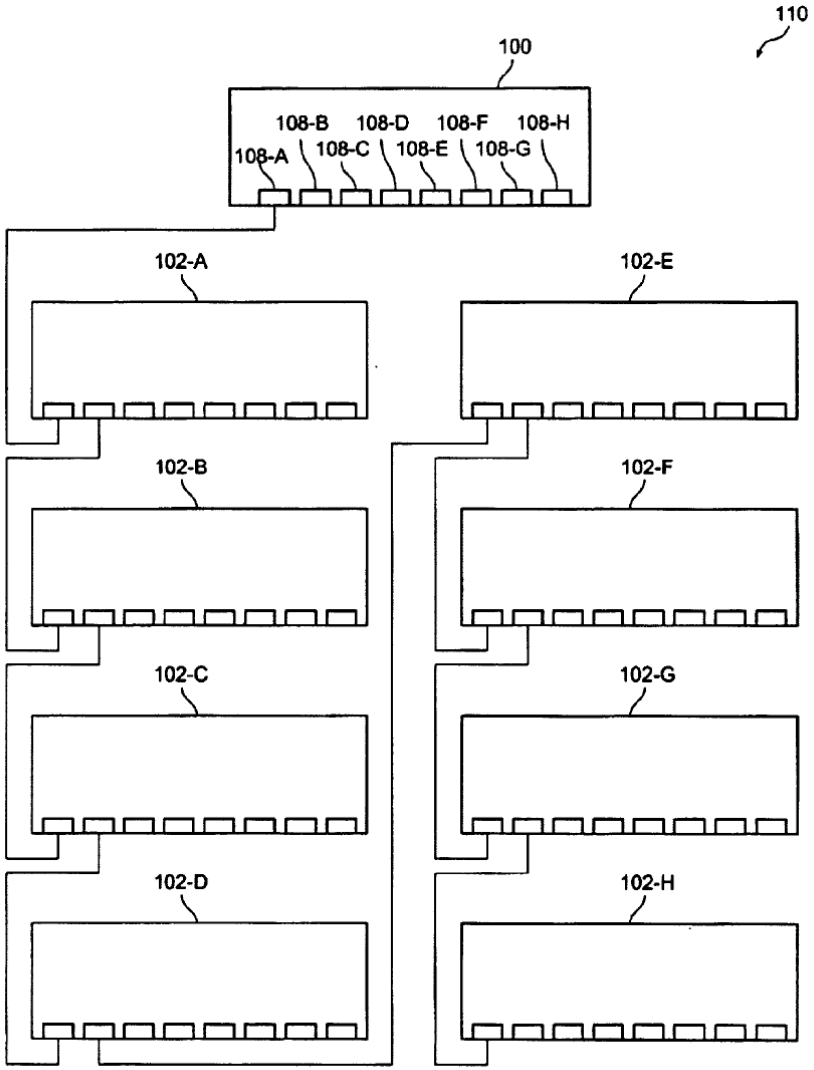
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>***FIG. 17***<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

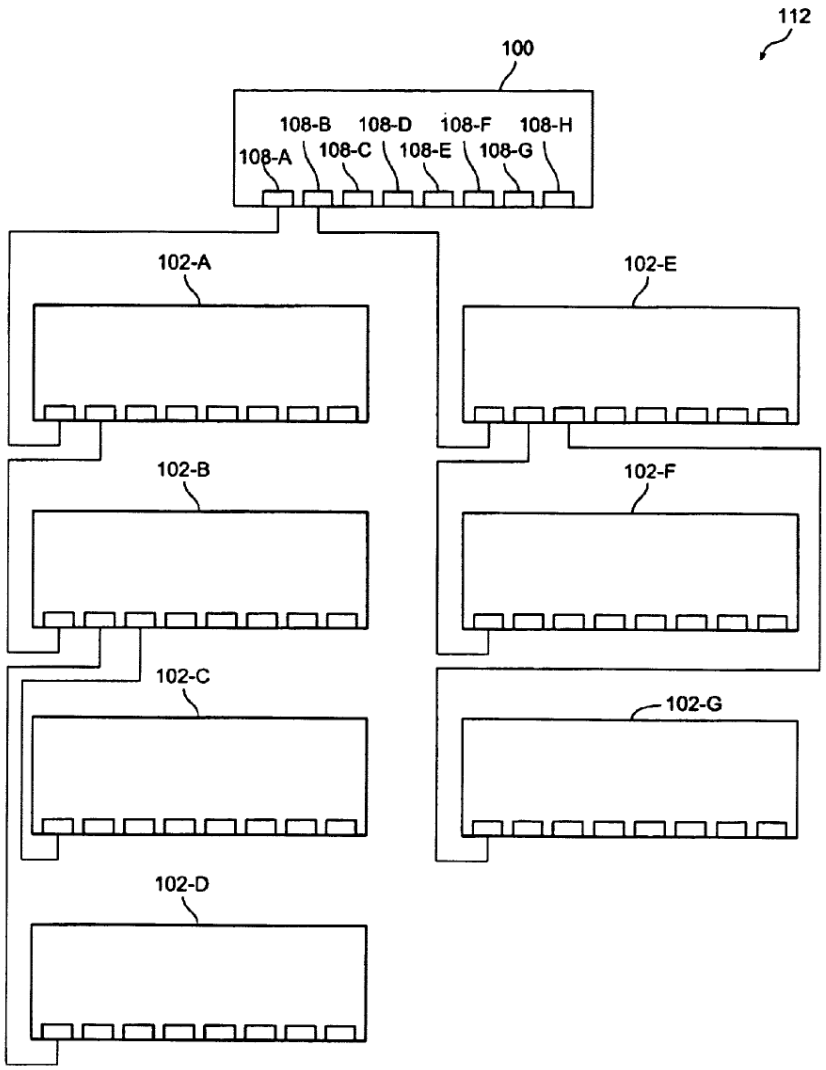
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10.<br><br>"Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

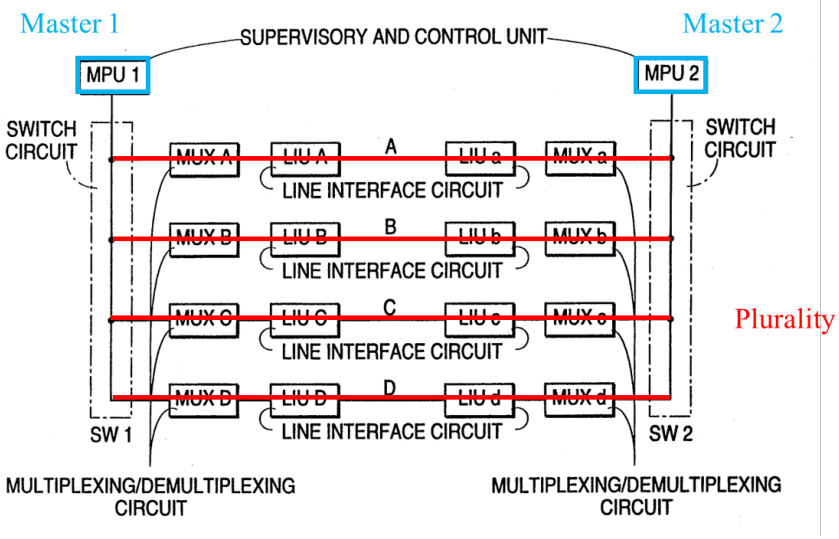
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

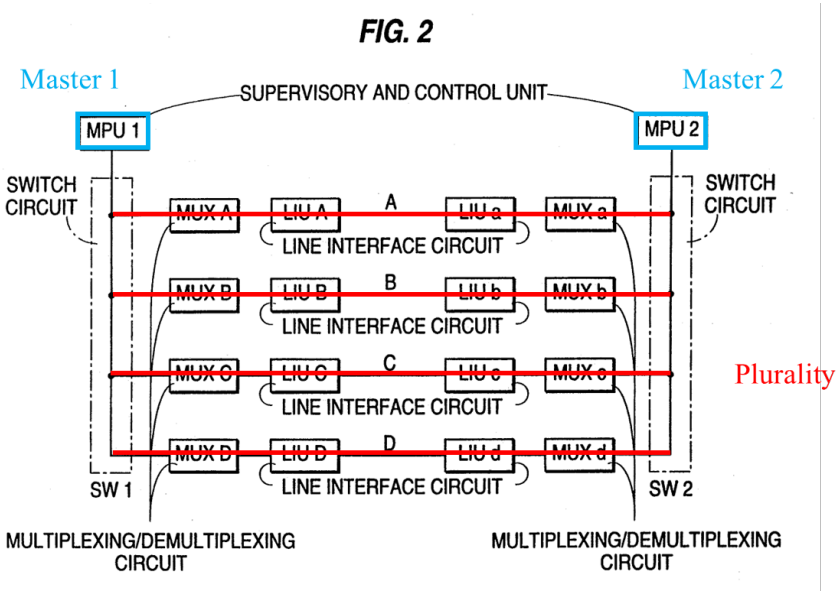
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 9[c] | each slave unit comprising one or more ports to respective subscriber lines; and | The Reference discloses each slave unit comprising one or more ports to respective subscriber lines.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[c]. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| 9[d] | a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween, | The Reference discloses a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling and Sugawara.<br><br>*See supra* at 1[d].<br><br>Cisco continues to make innovative contributions to the area of redundant stacked switch technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

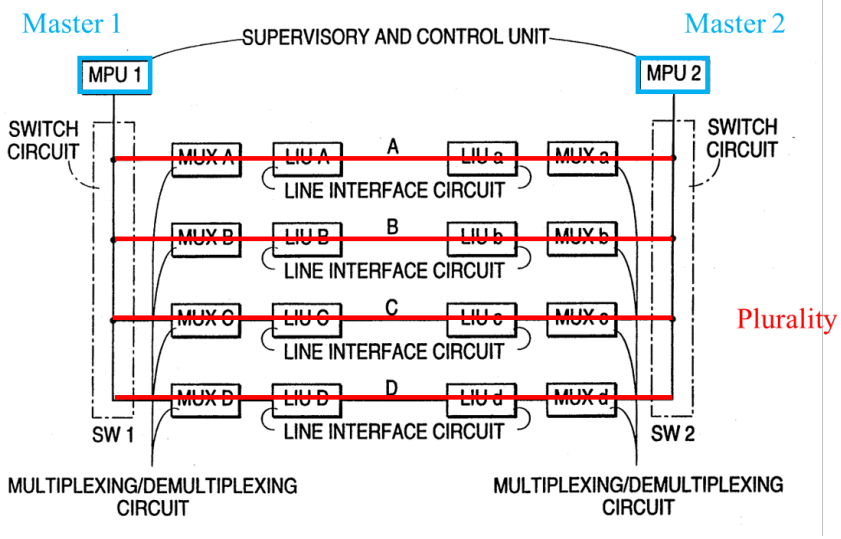
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotated).<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |
| 9[e] | each daisy chain comprising at least a first slave unit connected by one of the physical interface lines to the first master unit and a last slave unit connected by another of the physical interface lines to the second master unit, wherein each of the first and second master units comprises: | The Reference discloses each daisy chain comprising at least a first slave unit connected by one of the physical interface lines to the first master unit and a last slave unit connected by another of the physical interface lines to the second master unit, wherein each of the first and second master units comprises.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[e].<br><br>Below are examples of such references. |

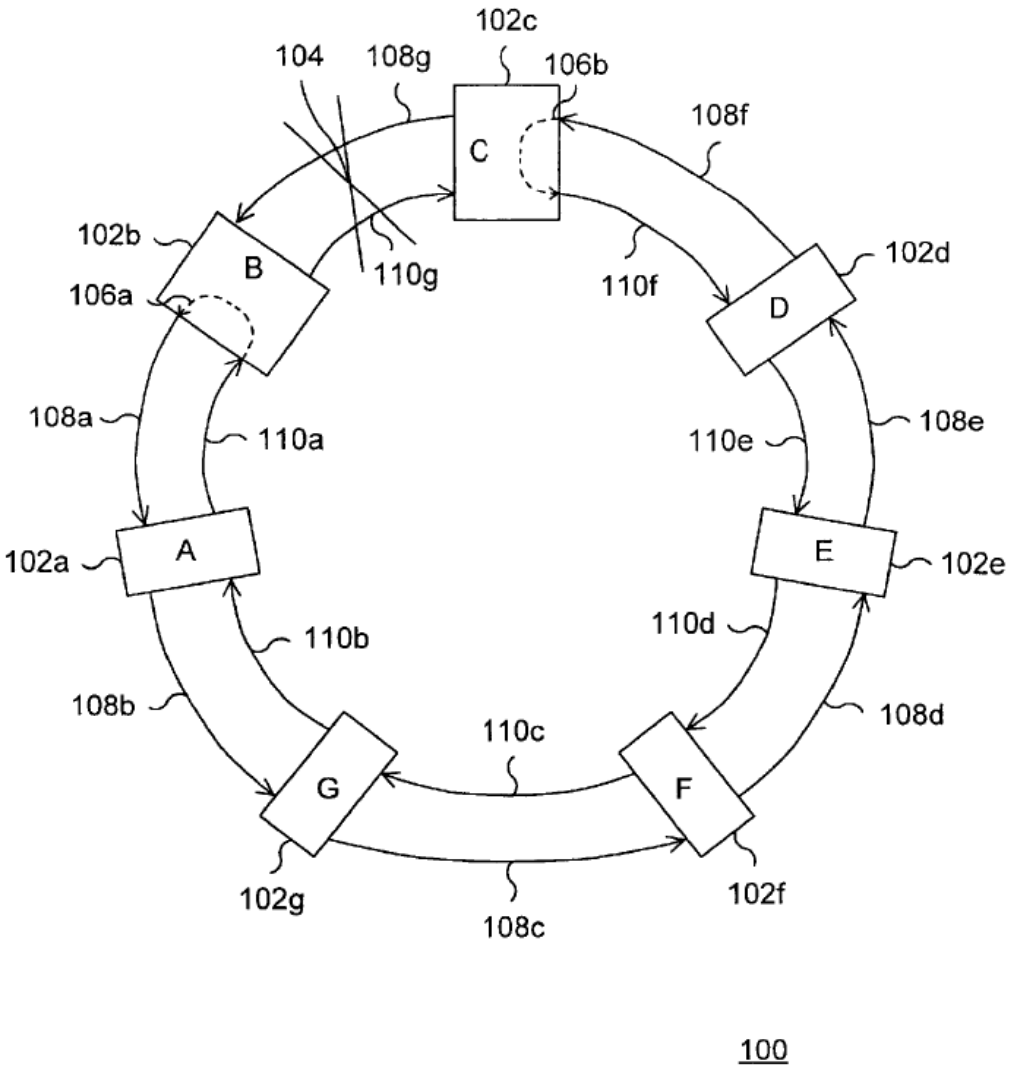
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | **Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided.").<br>  <br><br>FIG. 2 (annotation added)<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl |

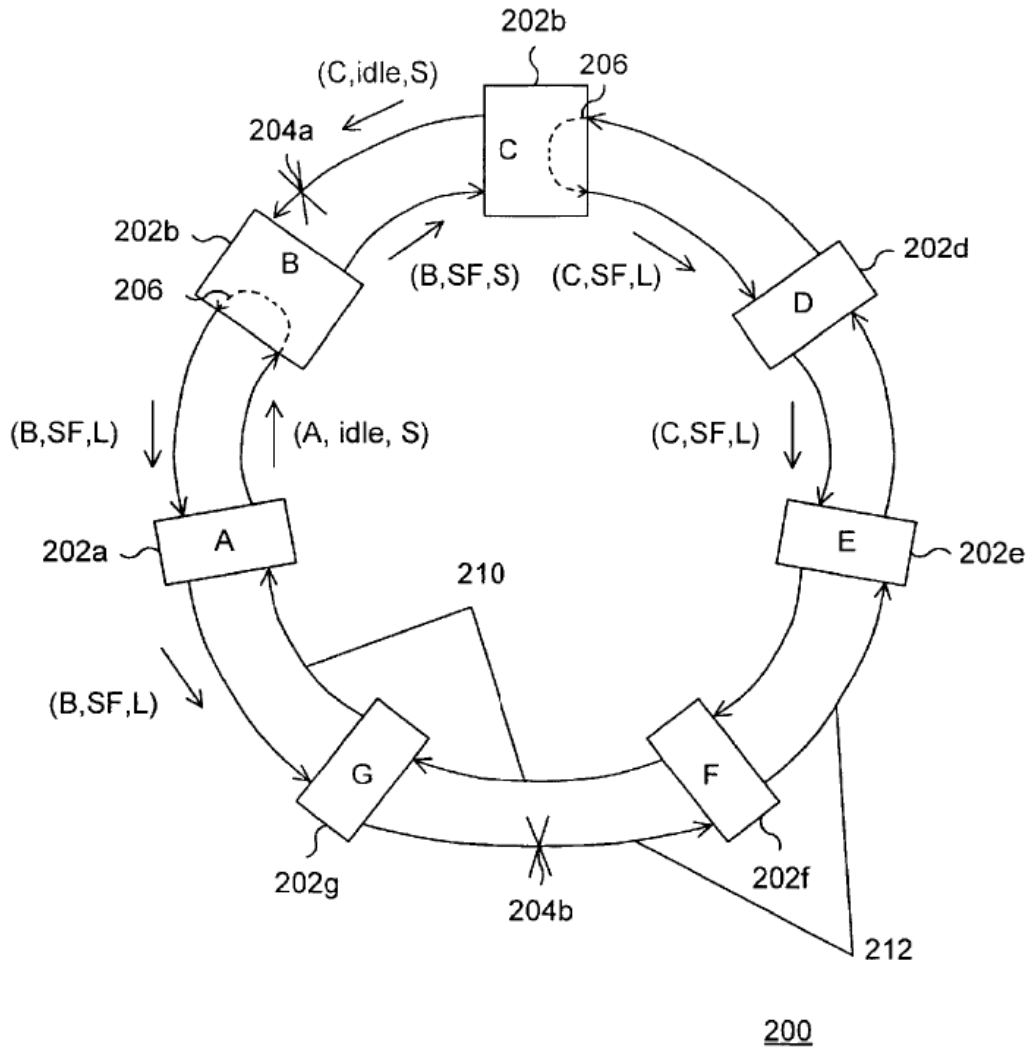
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |
| 9[f] | a switch, configured to route data packets between the respective physical interface and the one or more daisy chains, and | The Reference discloses a switch, configured to route data packets between the respective physical interface and the one or more daisy chains.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

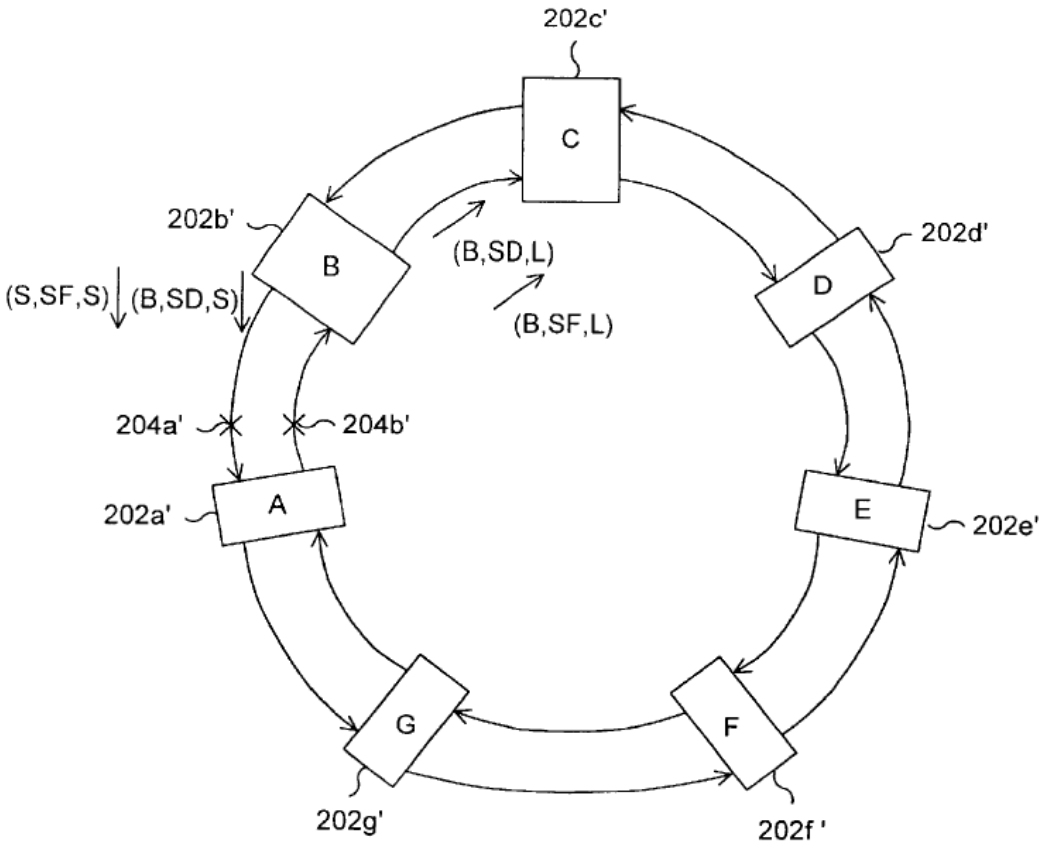
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| 9[g] | a pre-switch, which in the event of a fault at a location in one of the daisy chains, re-routes at least a portion of the data packets exchanged with one or more of the slaves in the daisy chain in which the fault has occurred through another one of the daisy chains. | The Reference discloses a pre-switch, which in the event of a fault at a location in one of the daisy chains, re-routes at least a portion of the data packets exchanged with one or more of the slaves in the daisy chain in which the fault has occurred through another one of the daisy chains.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco commercialized and patented technology relating to monitoring, detecting, and resolving faults without requiring a network reconfiguration *before* Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Daruwalla<br>• Nederveen<br>• Slater '421<br>• Petersen<br><br>Below are examples of such references.<br><br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotation added)<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.")<br><br>**Daruwalla discloses**:<br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. The present invention provides a protection protocol which simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol will appropriately remove a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, would remove protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, Abstract. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 1. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 2. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 11.<br><br>"The present invention relates to computer networks. In particular, the present invention relates to a system and method for providing a protection protocol for fault recovery for a two line bi-directional ring network." Daruwalla, 1:8-11. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "FIG. 1 shows an example of a two line bi-directional ring network. The ring network 100 is shown to include nodes 102 a-102 g. Each node is typically a computer with embedded processors and at least one network connection. Each node 102 a-102 g is shown to be bidirectionally coupled to two neighboring nodes 102 a-102 g via an inner connection ring 110 a-110 g and an outer connection ring 108 a-108 g. For instance, node 102 a is bidirectionally coupled to nodes 102 b and 102 g. The example of FIG. 1 also shows a problem 104 in the connection between node 102 b and node 102 c. When a problem is detected (such as a bi-directional line cut), the connection between nodes 102 b and 102 d wraps back upon itself, as shown by wraps 106 a and 106 b. In this manner, the connection problem 104 can be avoided." Daruwalla, 1:17-30.<br><br>"In a conventional SONET network, each message sent by a sending node to a receiving node typically needs the identification and location of the receiving node to arrive at the proper destination. Accordingly, manual configuration is typically needed in each node to store the identity and location of each other node in the ring network in order to provide for communication between the nodes in the network." Daruwalla, 1:31-44.<br><br>"In summary, for the protection mechanism to operate, each node needs to know the current ring map (current ring topology). What is needed is a system and method for providing fault recovery for two line bi-directional ring network that minimizes the need to keep track of other nodes in the ring network. Preferably, the system would not require reconfiguration of an internal map of the network when a new node is added to, or existing nodes are removed from the network. The present invention addresses such a need." Daruwalla, 2:23-31.<br><br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. According to the embodiment, when the problem is identified, the node sends a message identifying the problem to a second neighbor which is located at least one node away from the problem. The second neighbor then forwards the message to a third neighbor, unless the second neighbor is dealing with a situation that is higher in a hierarchy of situations than the problem described |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | in the message by the original node. In general, if the second neighbor's situation has a higher priority than the situation described by the original node, then the message is ignored and not forwarded. If, however, the message sent by the original node describes a situation with a higher priority than the situation being dealt with by the second neighbor, then, in general, the second neighbor's situation is ignored, at least for the moment, and the original node's message is forwarded to the next neighbor. In general, a higher priority request preempts a lower priority request within the ring. Exceptions are noted as rules of the protection protocol." Daruwalla, 2:35-58.<br><br>"The present invention provides a protection protocol that simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol automatically appropriately removes a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, removes protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, 2:59-3:3.<br><br>"A method according to an embodiment of the present invention for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:4-12.<br><br>"In another aspect of an embodiment of the present invention, a method for adding a new node to a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node prior to an addition of the new node; initiating a fault recovery procedure when the second node receives the first message; receiving a second |

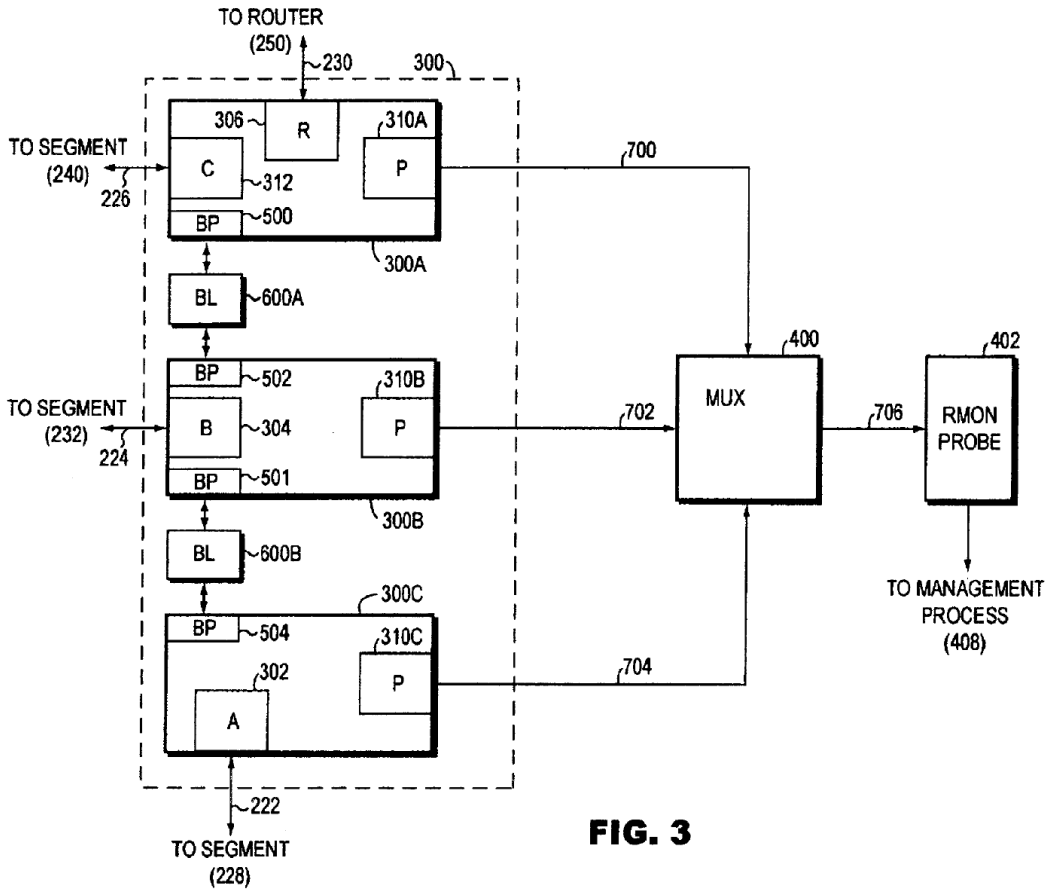
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | message from the new node; and entering an idle state when the second message is received." Daruwalla, 3:13-24.<br><br>"In yet another aspect of an embodiment of the present invention, a system for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The system comprises means for detecting a situation by a first node, wherein the first node is one of the plurality of nodes; means for sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and means for initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:25-35.<br><br>"FIG. 2 is block diagram of a ring network utilizing a protection protocol according to an embodiment of the present invention." Daruwalla, 3:40-42.<br><br>"FIGS. 4-6 are flow diagrams illustrating various rules within the protection protocol according to an embodiment of the present invention." Daruwalla, 3:46-48.<br><br>"FIGS. 8-12 are flow diagrams and a system diagram illustrating further rules of the protection protocol according to an embodiment of the present invention." Daruwalla, 3:52-54.<br><br>"FIG. 2 is a block diagram showing a ring network system utilizing a method of fault recovery according to an embodiment of the present invention. The ring network 200 is shown to include nodes 202 a-202 g. The nodes 202 a-202 g are shown to be coupled via an inner ring 210 in which the data flows in one direction, such as a clockwise direction. Additionally, the nodes 202 a-202 g are also shown to be coupled by an outer ring 212 in which data can flow in the opposite direction to the inner ring 210, such as in a counter-clockwise direction. The ring network 200 is shown to have a situation 204 a that requires protection, such as a ring wrap 206." Daruwalla, 5:35-45. |

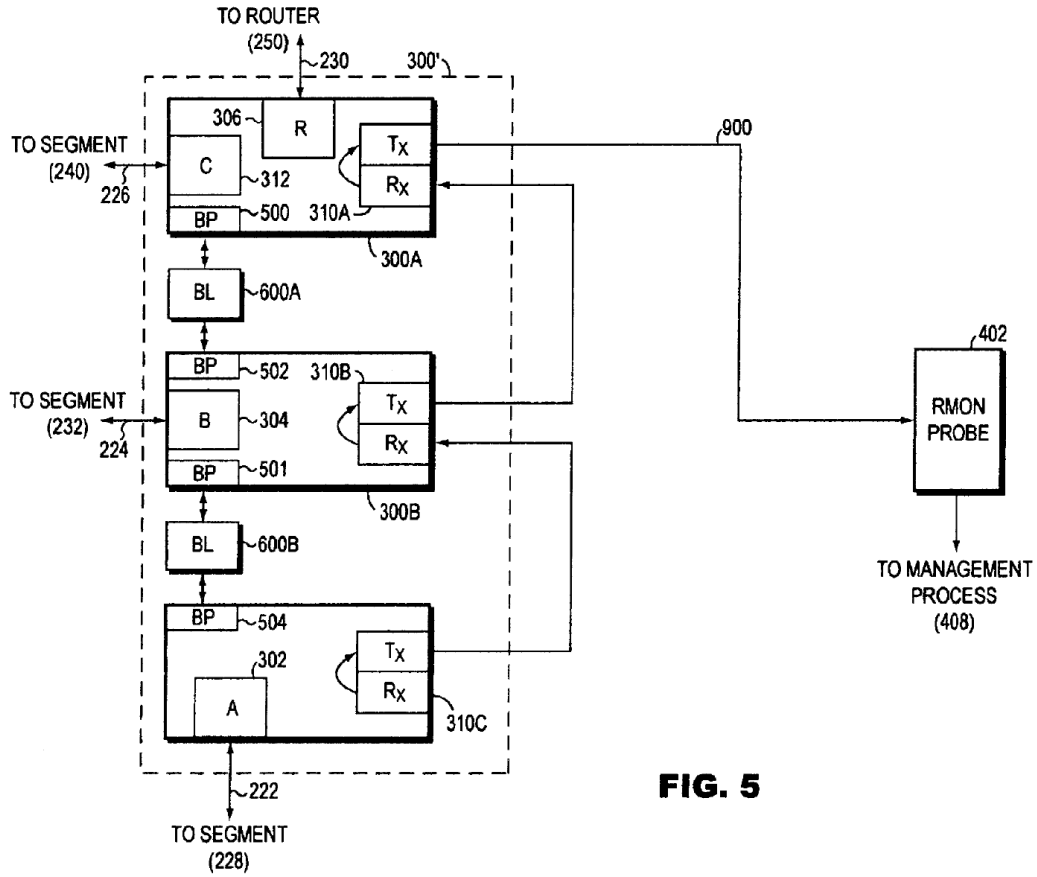
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "FIG. 4 is a flow diagram of an example of a method according to an embodiment of the present invention implied by Rules 1-22. An APS packet is received via step 400. It is determined whether the APS packet has been sent along a long path via step 402. If the packet was not sent via a long path, then the APS packet is not forwarded via step 406. Accordingly, if the APS packet was sent via the short path, then the packet is not forwarded via step 406. If, however, the packet was sent through the long path via step 402, then the APS packet may be forwarded via step 404. Note that for this example of Rule (1), it is assumed that the long path message does not have to pass through a wrapped connection in order to be forwarded. Otherwise, if the long path message needs to pass through a wrapped connection in order to be forwarded, then the message will simply not be forwarded." Daruwalla, 6:21-36.<br><br>"FIG. 6 is a flow diagram illustrating Rules 4 and 5. A node detects a problem between the node and a first neighbor via step 600. The node performs a wrap away from the side on which the problem exists via step 602. A short path message is then sent to the first neighbor informing it of the problem via step 604. Additionally, a long path message is also sent to a second neighbor informing the second neighbor of the problem via step 604. The first neighbor then performs a wrap away from the side of the problem via step 606. The first neighbor also sends an IDLE message, indicating a wrapped status, on a short path to the node that detected the problem via step 608. This message is sent across the failed span. Note that IDLE messages do not get wrapped and are sent across failed spans if possible. Additionally, the first neighbor also sends a message on a long path toward the side without the problem via step 608." Daruwalla, 6:64-11.<br><br>"An example of the method described in FIG. 6 can be seen in FIG. 2. Node 202 b has detected a problem 204 a and performs a wrap 206 on the side on which the problem exists. Node 202 b also sends a short path message to the neighbor on the other side of the problem 204 a, which is node 202 c. Node 202 b also sends a long path message to its other neighbor node 202 a informing it of the problem. Node 202 c performs a wrap 206 on the side of the problem and sends an IDLE message on a short path to node 202 b. Node 202 c also sent a message on a long path toward the side without the problem to its neighbor 202 d." Daruwalla, 7:12-21. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "FIG. 7 lists the hierarchy of priorities of Rule (8). For ease of reference, the hierarchy is separated into Class I-III. Class I is the highest priority, while Class III is the lowest priority. An example of a highest priority message in Class I is lockout. Lockout is an order stating that the ring network is not to wrap under any circumstances." Daruwalla, 7:22-26.<br><br>"Examples of the next priority listed in Class II are forced switch and signal fail. Forced switch indicates that the ring network is configured to wrap at the point of the forced switch. Signal fail is a situation where either two nodes cannot communicate with each other, or one node cannot hear the other node. An example of a signal fail is a physical break in the communication lines between two nodes." Daruwalla, 7:27-33.<br><br>"Note that members of Class II can co-exist (Rule 9). For example, multiple forced switches and signal fails can co-exist. Additionally, members of Class I can co-exist (Rule 10). For example, multiple lockouts in a single ring network can co-exist. However, situations in Class III cannot co-exist with other situations (Rule 11). For example, a signal degrade cannot co-exist with a wait-to-restore." Daruwalla, 7:52-58.<br><br>"When there are multiple requests of the same priority within Class III, the first request to complete a long path signaling will take priority (Rule 13). For example, if there are two signal degrades located on the same ring network, then the first signal degrade which completes the long path signaling will take priority over the other signal degrade. By not allowing members of Class III to co-exist with one another, partitioning of the ring network is avoided." Daruwalla, 7:59-65.<br><br>"In case of two equal requests within Class III on both inner and outer rings of the ring network, the tie is broken by choosing a request on one of the rings, such as the outer ring request (Rule 14). For example, if a signal degrade occurs both on the inner and outer rings, then a request on a predetermined ring, such as the outer ring, will take priority over the other requests." Daruwalla, 7:66-8:5. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "FIG. 8 is a flow diagram illustrating Rules (9), (10), (11), (13), and (15). Note that the flow diagram described in FIG. 8 is merely an example of one way in which the rules of the method according to the embodiment of the present invention can be executed. For example, the determination of whether the long path message is a Class I request via step 802 or a Class II request via step 810 can be in reverse order." Daruwalla, 8:6-11.<br><br>"A wrapped node receives a long path message via step 800. It is then determined if the long path message is a Class I request via step 802. The classes used in FIG. 8 are meant to correspond with the example of classes defined in FIG. 7. If the long path message is a Class I request, then it is determined if a local situation also has a Class I request via step 804. A local situation includes scenarios such as when a node detects a situation or problem, or when a node is made aware of a problem or situation via a short path message from its neighbor. If a local situation is not a Class I request via step 804, then any existing local wraps are unwrapped and the long path message is forwarded via step 806. If, however, a local situation is a Class I request via step 804, then the connections are already unwrapped or was never wrapped, and the long path message is forwarded via step 808." Daruwalla, 8:12-26.<br><br>"FIG. 12 is a flow diagram illustrating rules (20) and (21) of the method according to the embodiment of the present invention. A wrapped node determines that a problem has been cleared via step 1200. It then enters a wait-to-restore state via step 1202. It is then determined if its neighbor is the same neighbor as previously noted via step 1204. The node can save the source of a short path message at the time of wrap initiation to note the identity of its neighbor. If the current neighbor is not the same as the previous neighbor via step 1204, then an IDLE state is entered via step 1206. If, however, the current neighbor is the same as the previous neighbor via step 1204, then it is determined whether a pre-determined wait-to-restore time has expired via step 1208. Once the pre-determined wait-to-restore time has expired, then the node enters an IDLE state via step 1206." Daruwalla, 12:60-13:6.<br><br>"A method and system for fault recovery for a two line bi-directional network has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor." Daruwalla, 13:7-19. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | **Nederveen discloses**:<br>"A technique for use in gathering network activity-related information from cascaded network switches is provided. Using this technique, the information can be gathered without substantially reducing performance of the cascaded switches. In one embodiment, a single remote monitoring probe is connected via respective connections to each of the switches so as to receive the information from the switches. In another embodiment, only one of the switches is connected to the probe, and the other switches transmit their respective portions of the information to the switch connected to probe. The switch connected to the probe provides these portions of the information, as well as, any of its respective activity-related information to the probe. In this latter embodiment, the switches may be connected by dedicated connections and switch ports that are used solely for communicating the activity-related information." Nederveen, Abstract. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>**FIG. 3**<br><br>Nederveen, FIG. 3. |

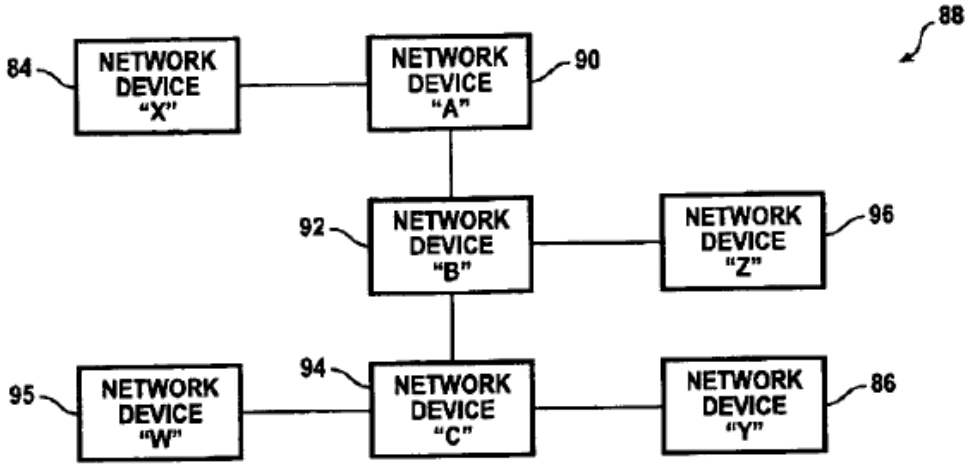
| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>**FIG. 5**<br><br>Nederveen, FIG. 5.<br><br>"Thus, it would be desirable to provide a stacked switch monitoring technique that permits efficient offloading of raw data processing from the stacked switches, requires only a minimal number of specialized network entities to gather and process such raw data, and does not result in substantial degradation of stacked switch performance." Nederveen, 4:38-43. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "Accordingly, the present invention provides a technique for remote monitoring of a switch network that overcomes the aforesaid and other disadvantages and drawbacks of the prior art. More specifically, in one aspect of the present invention, a technique is provided for gathering information that may be useful in network management (e.g., switch port activity-related information), from switches in the network that are in a stacked configuration. The information is gathered from the stacked switches by a single network entity (e.g., an SNMP remote monitoring probe) in such a way that it does not substantially degrade the performance of the switches. This is accomplished, in one embodiment of the technique of the present invention, by connecting the switches via respective connections to a multiplexer that selectively connects the switches, according to an arbitration scheme, to the single network entity. The entity gathers respective portions of the information from switches when it is connected to the switches by the multiplexer. The information gathered by the entity may be provided to another network entity (e.g., an SNMP management node) in order to permit the other entity to use that information in managing the network." Nederveen, 4:46-67.<br><br>"In another embodiment of the technique of the present invention, only one of the switches is connected to the single information gathering entity. The switches that are not connected to the entity transmit, via respective dedicated ports and connections (i.e., ports and connections that are used solely for network information gathering activities), their respective portions of the information to the switch that is connected to the entity. The switch that is connected to the entity transmits, via a respective dedicated port and connection, the information received from the other switches, as well as, its own information to the entity." Nederveen, 5:1-11.<br><br>"FIG. 3 is a schematic, functional block diagram illustrating in greater detail the construction of the stacked switch network shown in FIG. 2." Nederveen, 5:26-28.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network configured to employ another embodiment of the present invention." Nederveen, 5:32-34. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "FIGS. 2-5 illustrate features of a computer network 200 wherein embodiments of the present invention may be advantageously practiced. Network 200 comprises a stacked switch network 300 which interconnects a plurality of network segments 228, 232, 240, and 251. Each segment 228, 232, 240 comprises one or more local area networks having computer endstations (not shown). Segment 251 is a network router segment that comprises network router 250. Each segment 228, 232, 240 is coupled via a respective communications link 222, 224, 226 to a respective port 302 (i.e., port A), 304 (i.e., port B), 312 (i.e., port C) of the switch network 300. Likewise, the router 250 of router segment 251 is coupled via a respective trunk line 230 to router port 306 (i.e., port R)." Nederveen, 5:46-59.<br><br>"Stacked switch network 300 comprises a plurality of data network switches 300A, 300B, 300C (e.g., Catalyst 3900™ series switches of the type commercially available from the Assignee of the subject application) coupled together via conventional stack link bus connection logic 600A, 600B. More specifically, logic 600A couples a stack link bus port and associated logic 500 in switch 300A to a stack link bus link port and associated logic 502 in switch 300B. Similarly, logic 600B couples another stack link bus port and associated logic 501 in switch 300B to a stack link bus port and associated logic 504 in switch 300C. It should be understood that although, as is shown in FIG. 3, switches 300A and 300B, and switches 300B and 300C, may be coupled serially together by separate respective logic elements 600A, 600B, each of the switches 300A, 300B, 300C may be coupled together via a single respective stack link bus port in the switch to a single stack link bus connection logic block (not shown, e.g., of the type that is commercially available under the tradename Catalyst Matrix™ from the Assignee of the subject application). Further alternatively, depending upon the particular design and functionality of the ports 500, 501, 502, and 504, and the control and forwarding logic (whose operation will be described more fully below) in the switches 300A, 300B, 300C, the circuitry in logic 600A, 600B may instead be comprised in the ports 500, 501, 502, and 504 and/or control and forwarding logic, and therefore, in this alternative configuration, the logic 600A, 600B in the network 300 may be replaced by simple connection means (e.g., cable connectors)." Nederveen, 6:29-57. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "Each switch 300A, 300B, 300C includes a respective internal bus (e.g., element 800 in switch 300C) that is coupled via at least one stack link bus port and associated interface logic (e.g., 504 in switch 300C) to external stack link bus connection logic (e.g., element 600B in switch 300C). Each switch 300A, 300B, 300C also includes respective programmable control and forwarding logic (e.g., element 802 in switch 300C) comprising processing, memory, and other circuitry for storing and learning configuration information (e.g., source and destination MAC addresses of messages received by the switch, switch bridging table, switch segments' spanning tree and virtual local area network information, etc.), and for providing appropriate commands to other elements (e.g., the switch ports) to cause data messages received by the switch to be forwarded to appropriate network segments coupled to the switch based upon this configuration information. In each switch, the switch's port logic circuitry (e.g., port A logic 302 and port P logic 310C in switch 300C) and control and forwarding logic are coupled to each other via that switch's respective internal bus. The stack link bus port and associated logic in each switch 300A, 300B, 300C may comprise a Catalyst™ stack port line interface card (commercially available from the Assignee of the subject application) inserted into a bus expansion slot (not shown) in the switch. Although not shown in the Figures for purposes of clarity of illustration, each switch 300A, 300B, 300C in network 300 typically will include tens or hundreds of ports coupled to network segments." Nederveen, 6:58-7:19.<br><br>"The control and forwarding logic and stack link bus port and associated logic in each switch, and the logic 600A, 600B, are configured to together implement conventional techniques for permitting the switches 300A, 300B, 300C to function together as a single logical/virtual switch. More specifically, when configured in the stacked arrangement 300, after the switches 300A, 300B, 300C and logic 600A, 600B are initially activated, they execute initial power-on self-diagnostics, and thereafter, enter a "stack discovery" mode of operation." Nederveen, 7:20-29.<br><br>"In the stack discovery mode of operation, the control and forwarding logic in each switch 300A, 300B, 300C first "senses" that its switch is coupled to logic 600A and/or 600B, and then determines the particular configuration of the stacked switch network 300, using suitable conventional autosensing/autoconfiguration techniques. The control and 161orwarding logic in the switches 300A, 300B, 300C then assigns to the switches respective unique |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | identification numbers (e.g., based upon unique identification numbers of respective ports of the logic 600A, 600B to which the switches are coupled)." Nederveen, 7:30-40.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network 300′ configured to employ another embodiment of the present invention. It should be understood that unless specifically stated to the contrary, the structure and operation of the network 300′ are substantially the same as the structure and operation of network 300. In network 300′, each of the dedicated ports 310A, 310B, 310C comprises a respective transmit portion and receive portion, referenced in FIG. 5 as RX and TX, respectively." Nederveen, 11:20-29.<br><br>**Slater '421 discloses**:<br>"A method and apparatus for discovering paths to other network devices includes a protocol and network management application that can be executed on network devices. The Ethernet protocol is used to detects paths to other network devices, knowing only the Ethernet address of the destination. A discovery protocol is extended to add hop probe and hop probe reply Type-Length-Value fields in a variable-length list. The hop probe fields contain a hop count, a destination Ethernet address, and a source Ethernet address. When a hop probe is received by a network device, the hop count field is decremented by one and the hop probe is forwarded. Packet received with a hop count of one are not forwarded and a hop probe reply is sent back to the Ethernet source address of the hop probe. The hop probe reply fields contain a destination Etherned address and a source Ethernet address." Slater '421, Abstract. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br><br>Slater '421, FIG. 6.<br><br>"Partly as a result of the increased complexity of networks, network administrators must often troubleshoot problems with their network. Two classes of network problems often faced by network administrators are "reachability" problems and performance slowdowns. Reachability problems occur when one or more network devices cannot be accessed through a network, and can be caused by hardware or software failures, cabling problems, or any of several other types of difficult-to-diagnose problems that can occur in a network." Slater '421, 7:11-20.<br><br>"Some of the world's largest networks today rely on the TCP/IP suite of networking protocols. With a relatively small kit of basic tools, network administrators can learn much about an internetwork. "Ping" and "traceroute" commands, "show" commands, and "debug" commands (all of which are typically available via the basic management interface on a network device) form the core of the network administrator's internetwork toolkit. Ping and |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | traceroute commands can be useful tools in determining where failures are occurring, but they are cumbersome to use, and require knowledge of the IP address or host name of the destination network device. The show commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data. Finally, debug commands provide clues about the status of a network device and other network devices directly or indirectly connected to it. Because debug commands can create performance slowdowns, they must be used with great care, and using the wrong debug command at the wrong time can exacerbate problems in already poorly performing networks." Slater '421, 7:55-8:8.<br><br>"Embodiments of the present invention as illustrated herein use the Cisco™ Discovery Protocol ("CDP") to automatically detect paths to specified network devices in Ethernet LANs. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task." Slater '421, 9:10-15.<br><br>"CDP is a media-independent device discovery protocol which can be used by a network administrator to view information about other network devices directly attached to a particular network device. In addition, network management applications can retrieve the device type and SNMP-agent address of neighboring network devices. This enables applications to send SNMP queries to neighboring devices. CDP thus allows network management applications to discover devices that are neighbors of already known devices, such as neighbors running lower-layer, transparent protocols." Slater '421, 9:16-26.<br><br>"It is to be understood that the present invention is not limited to devices that are compatible with CDP. CDP runs on all media that support the Subnetwork Access Protocol ("SNAP"), including LAN and Frame Relay. CDP runs over the data link layer only. Each network device sends periodic messages to a multicast address and listens to the periodic messages sent by others in order to learn about neighboring devices and determine when their interfaces to the media go up or down. Each device also advertises at least one address at which it can receive SNMP messages. CDP messages, or "advertisements," contain holdtime information, which indicates the period of time a receiving device should hold CDP information from a neighbor before discarding it. With CDP, network management applications can learn the |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | device type and the SNMP-agent address of neighboring devices. This process enables applications to send SNMP queries to neighboring devices." Slater '421, 9:27-43.<br><br>"It should be noted that-normally, CDP packets according to aspects of the present invention are transmitted at regular intervals (e.g. once every 60 seconds). However, in embodiments of the present invention, when a Hop Probe or Hop Probe Reply needs to be forwarded by a network device, the network device is commanded to send a CDP packet immediately." Slater '421, 16:66-17:5.<br><br>"The present invention is much faster than the previous method that involved logging in to each intermediate network device, entering the "show cdp neighbors" command, and interpreting the output to find the next hop along the path to the destination network device. Also, the present invention allows individual users, such as network administrators, to execute a tool to manually discover paths through a network of Ethernet switches. The present invention can be used by network management software to automatically map the topology of clusters of network devices, such as Ethernet switches. Finally, the present invention is useful in loop detection. Enhancements to Spanning Tree and other bridge-level routing protocols can test proposed changes to switch topology prior to making them." Slater '421, 17:6-20.<br><br>**Petersen discloses**:<br>"Methods and apparatus for enabling communication between a source network device and one or more destination network devices are disclosed. A system enabling communication between a source network device and one or more destination network devices includes a switch and a ring interconnect. The switch is adapted for connecting to the source network device and the one or more destination network devices. More particularly, the switch is capable of storing data provided by the source network device and retrieving the data for the one or more destination network devices. The ring interconnect is adapted for connecting the source network device and the one or more destination network devices to one another. In addition, the ring interconnect is capable of passing one or more free slot symbols along the ring interconnect. Thus, the ring interconnect is capable of expanding when one or more of the free slot symbols are each replaced by a frame notify message indicating that the data has |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | been stored by the switch for retrieval by the one or more destination network devices." Petersen, Abstract. |

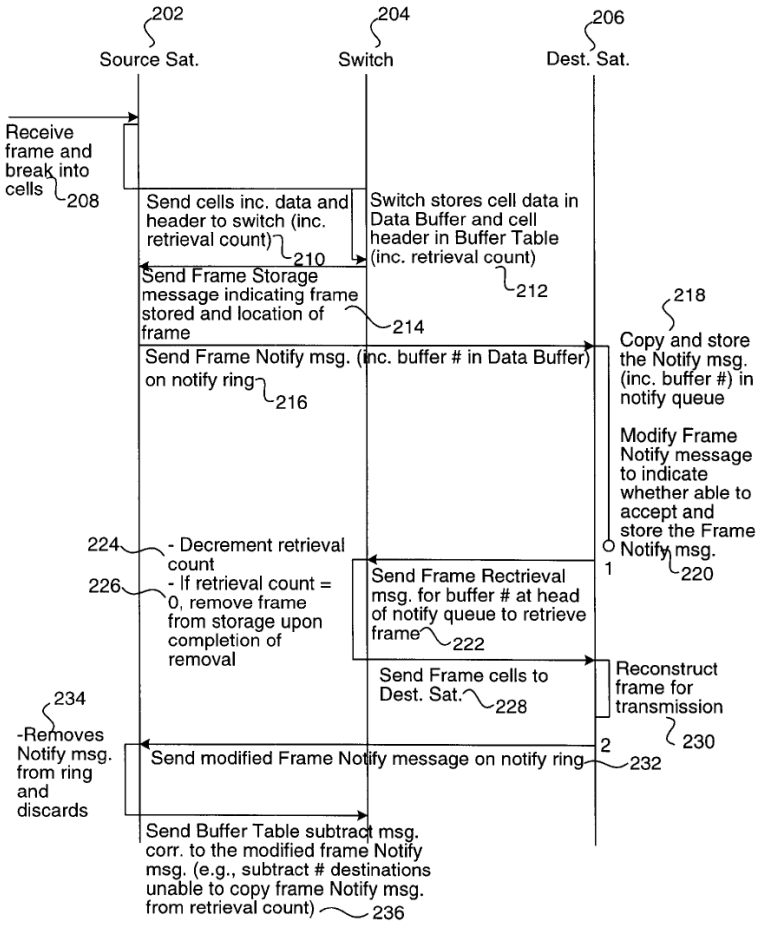



***FIG. 2***

Petersen, FIG. 2.

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "The present invention relates to a mixed topology data switching system that combines a radial interconnect with a ring interconnect. More particularly, the radial interconnect permits devices to store and retrieve data using a switch, while the ring interconnect permits devices along the ring interconnect to provide notification that data has been stored for retrieval, as well as provide feedback regarding the ability or inability to retrieve such data." Petersen, 1:34-41.<br><br>"In controlling the flow of network traffic through a switching system, it is often desirable to provide feedback to the source of the data. For instance, although a transmitting device, hereinafter referred to as a "source device," may transmit or forward data to a receiving device, hereinafter referred to as a "destination device," the destination device may be incapable of handling the data. In these circumstances, the source device is often unaware that the data was not accepted by the destination device, complicating switch management. Common solutions to the problem of switch traffic management have included ensuring that all intended destination devices are "ready to receive" prior to transmitting data on a ring or bus interconnect, or insisting that each intended destination device send an explicit acknowledgement back to the source device. Both of these approaches result in reduced efficiency of the interconnect scheme. By way of example, in a ring network, such acknowledgment is typically provided in the data frame being transmitted. As another example, in other interconnect schemes, each such device may send a separate acknowledgment, therefore adding to the traffic on the network. Accordingly, it would be desirable if a traffic management scheme were established which could provide such feedback to the source of the data while minimizing traffic management overhead." Petersen, 2:8-32.<br><br>"According to one embodiment, the present invention combines the use of two data transport methods: a point-to-point radial interconnect and a ring interconnect. The radial interconnect connects interface devices to each other through the services of a central switch device to permit the transport of data. Typically, a single interface has a single dedicated radial interconnect to the central switch. These interface devices are further connected to one another via a ring interconnect to convey retrieval notifications regarding forwarding of the data (by source devices) and receipt of the data (by destination devices)." Petersen, 2:36-46. |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | "Each radial interconnect provides a narrow, high baud-rate connection to convey to the actual data from and to the associated interface without being burdened by the unrelated traffic for the remaining interfaces in the system. This is accomplished through the use of a central switch device, which stores and retrieves data for the various interfaces in the system. As will be apparent from the following description, this architecture provides numerous advantages over a wide parallel bus or ring." Petersen, 2:47-55.<br><br>"The ring interconnect may be used to convey a "retrieval notification"(i.e., retrieval message) that may be observed by all potential retrieving interfaces. The retrieval notification notifies specific devices ("destination devices") or interfaces that one or more frames addressed to them are available from the switch device. Moreover, the ring interconnect permits each destination device to provide feedback to the source device letting the source know whether the destination has accepted the notification provided by the source device and therefore whether the destination can retrieve the data intended for it. The feedback is particularly useful in buffer management applications. In this manner, an efficient and flexible data transport and retrieval notification system that includes a feedback path to the source of the data is provided." Petersen, 2:56-3:3.<br><br>"FIG. 2 is a process flow diagram illustrating a method of providing network communication according to an embodiment of the invention." Petersen, 3:9-11.<br><br>"FIG. 2 is a process flow diagram illustrating in further detail a method of providing network communication in the above-described system according to an embodiment of the invention. As shown, process steps performed by a source device 202 are illustrated along an associated vertical line, steps performed by a switch 204 are illustrated along another vertical line, and steps performed by a destination device 206 are illustrated along still another vertical line." Petersen, 4:50-57.<br><br>"When the frame is stored by the switch 418, the source device preferably receives an acknowledgment that the data has been stored. Thus, to provide this feedback, a frame storage message (i.e., storage reply) is sent from the switch 418 on the channel 416 to the channel interface 414. The frame storage message is then provided to the notify ring interface as |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | shown at 430 and sent on the notify ring. Once this acknowledgment is received by the interface device 402, the designated destination devices may be notified via notify ring interface 424. As described above, a Frame Notify message may be sent via the notify ring interface 424 to the destination devices. More particularly, the Frame Notify message may identify one or more destination devices for the frame and specify the location of the frame to be retrieved. By way of example, the location of the frame to be retrieved by the destination devices may be designated by a buffer number 430. In addition, the destination devices for the frame may be specified in the Frame Notify message through a notify queue map 426. More particularly, the notify queue map 426 may specify a notify queue associated with a particular destination device. The notify queue may be expressly designated through the use of one or more bits as well as implied through the specification of a priority level for the data. The notify queue map 426 will be described in further detail with reference to FIG. 13. The notify ring interface 424 then creates a Frame Notify message including the notify queue map 426 and the buffer number 430 which is then sent on an outbound interface of the notify ring 432." Petersen, 7:55-8:16.<br><br>"As described above, the notify ring may be expanded to accommodate communication between interface devices. The communication between the interface devices and the switch is therefore performed on one or more channels rather than the notify ring. As a result, the flexibility of the notify ring does not effect the speed with which the interface devices may communicate with the switch. Thus, where a single port operates at a faster speed than the channels, multiple channels may be grouped together. In this manner, the speed with which the switch may communicate with the interface devices may be maximized." Petersen, 20:27-36.<br><br>"The present invention provides a mixed topology data switching system that combines a point-to-point radial interconnect with a ring interconnect to maximize the speed of network traffic. The radial interconnect provides a narrow, high baud-rate connection to convey the data traffic for just the interface in question, without being burdened by all of the unrelated traffic for the remaining interfaces in the system. At the same time, the ring interconnect permits retrieval notifications to be observed by all potential retrieving interfaces. The ring topology further permits each destination interface to provide feedback to the source |

| No. | '904 Patent Claim 9 | The Reference |
|---|---|---|
| | | interface, which is valuable for buffer management applications. Moreover, the point-to-point ring topology bus employs a variable latency access method that enables messages to be passed across the bus with low latency when the system is quiet and with increased latency when the system is busy. In addition, since control messaging around the ring interconnect and across the channel interconnects are embedded in the data stream, the number of pins required and manufacturing costs are reduced." Petersen, 20:38-57. |

| No. | '904 Patent Claim 10 | The Reference |
|---|---|---|
| 10 | Apparatus according to claim 9, wherein the pre-switch re-routes the data packets such that substantially no reconfiguration of the switch is required responsive to the fault. | The Reference discloses apparatus according to claim 9, wherein the pre-switch re-routes the data packets such that substantially no reconfiguration of the switch is required responsive to the fault.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| 11[preamble] | Network access apparatus, comprising: | The Reference discloses network access apparatus, comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| 11[a] | first and second master units, each comprising a physical interface to a packet-switched network; | The Reference discloses first and second master units, each comprising a physical interface to a packet-switched network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other documents) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626<br>• Duuvury '820<br><br>Cisco Catalyst Press Release discloses:<br><br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2. |

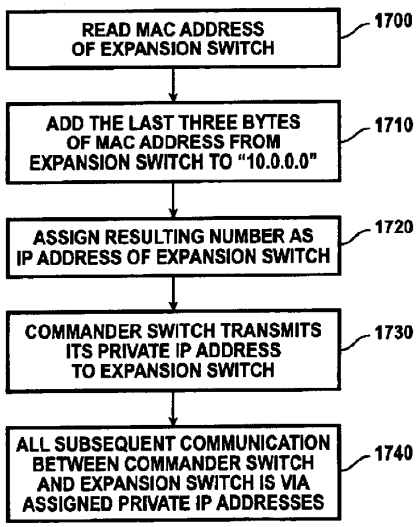
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2.<br><br>"With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3. |

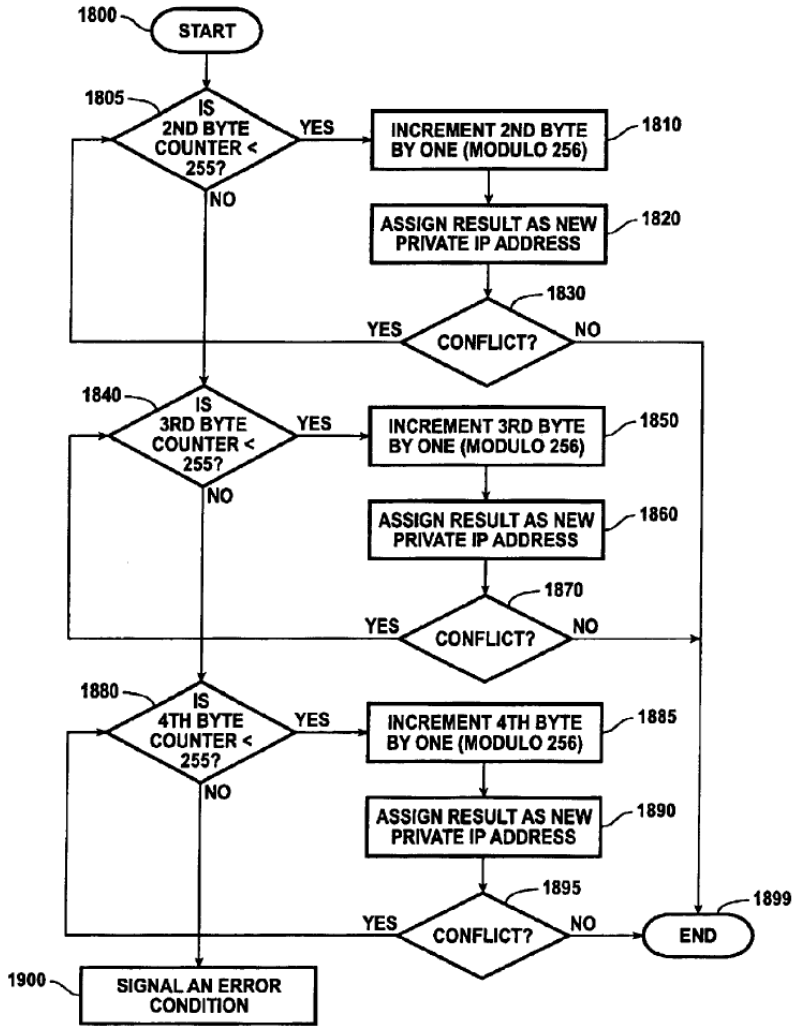
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract.<br><br>"If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

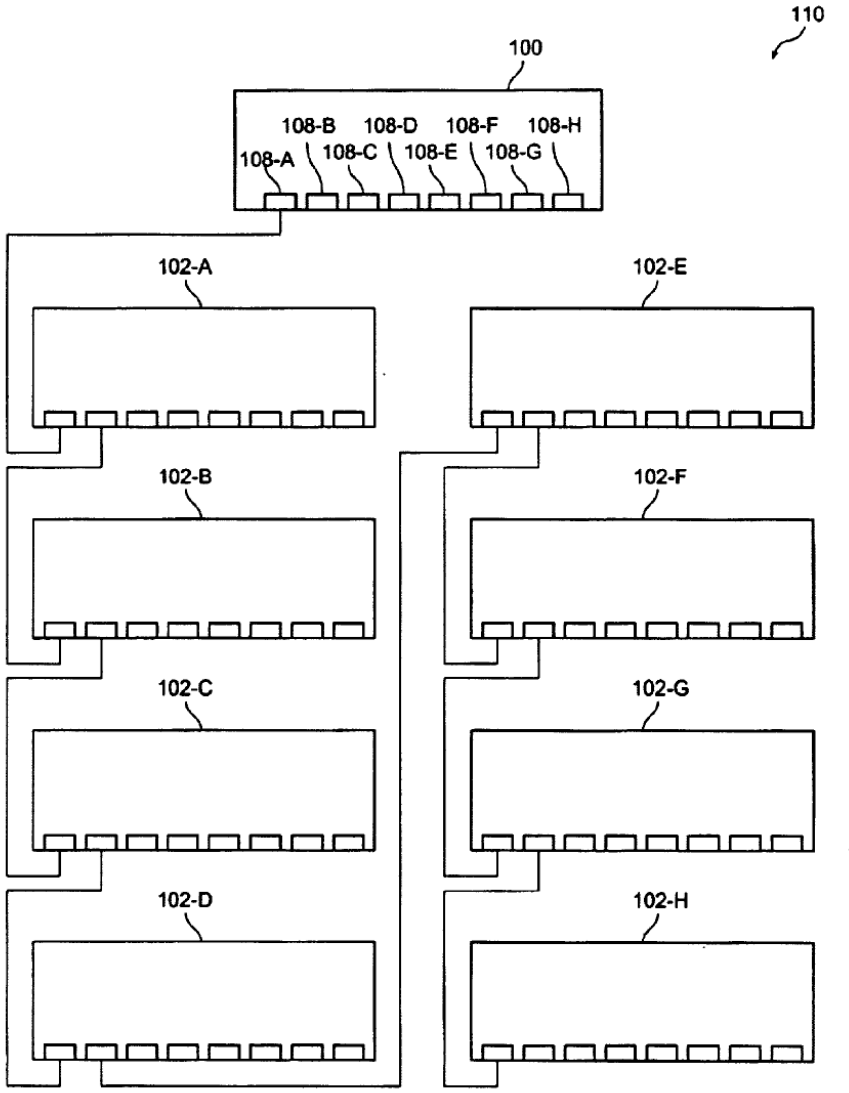
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

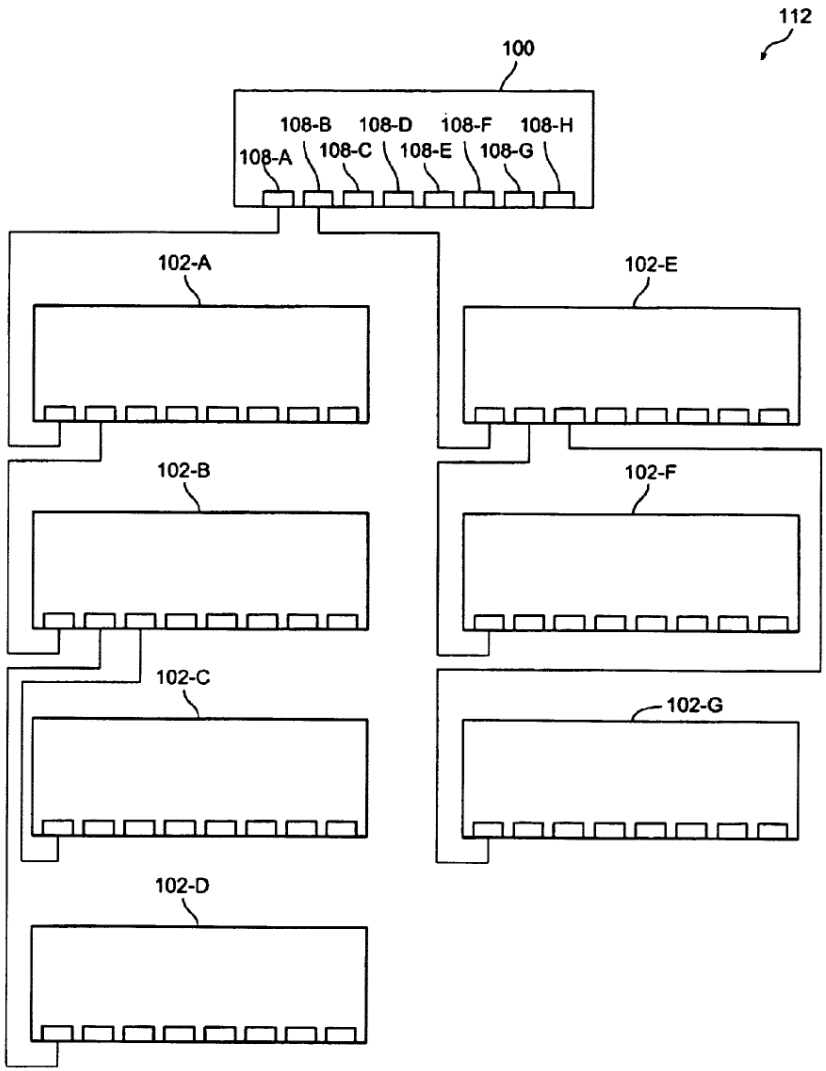
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 11[b] | a plurality of slave units, | The Reference discloses a plurality of slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433 |

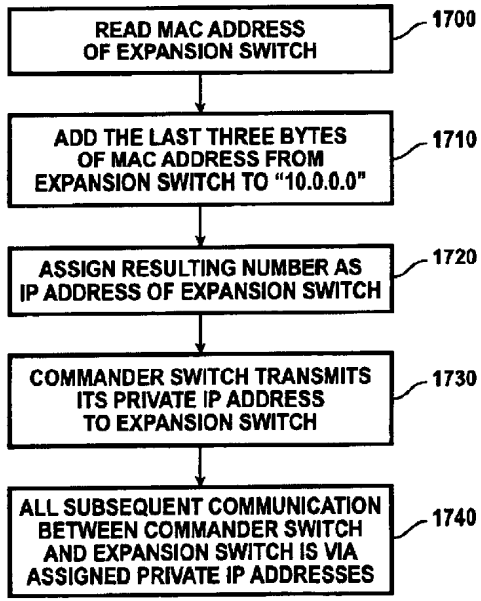
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | • Duvvury '626<br>• Duuvury '820<br><br>**<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2. |

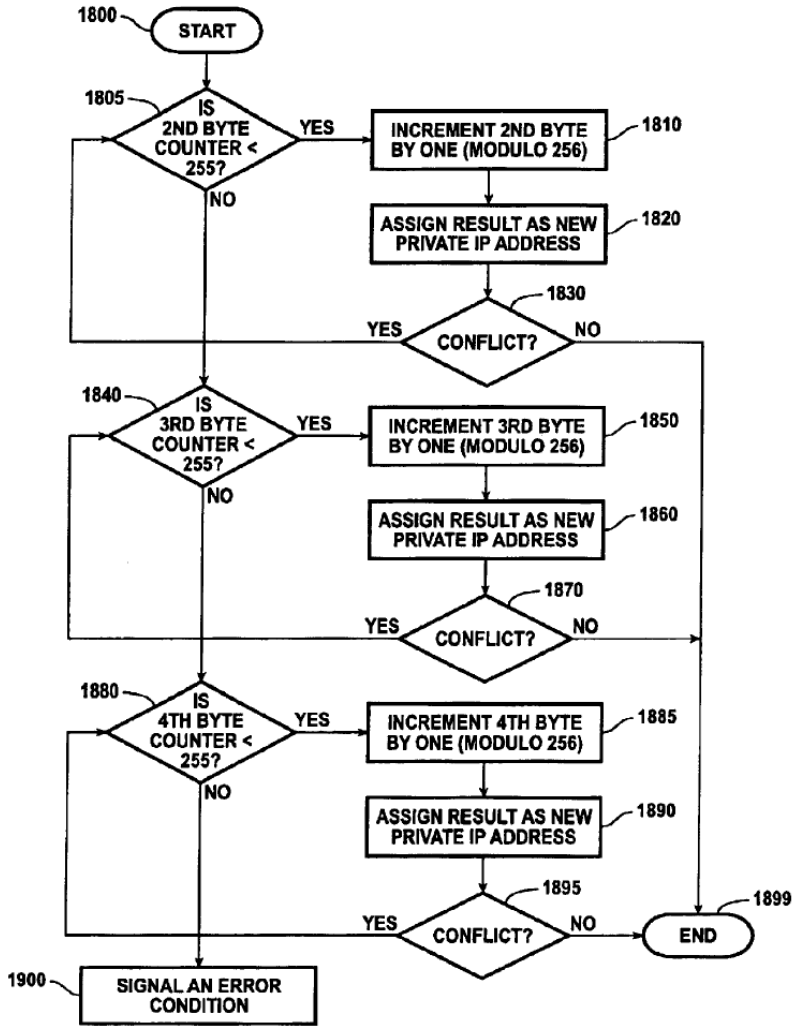
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

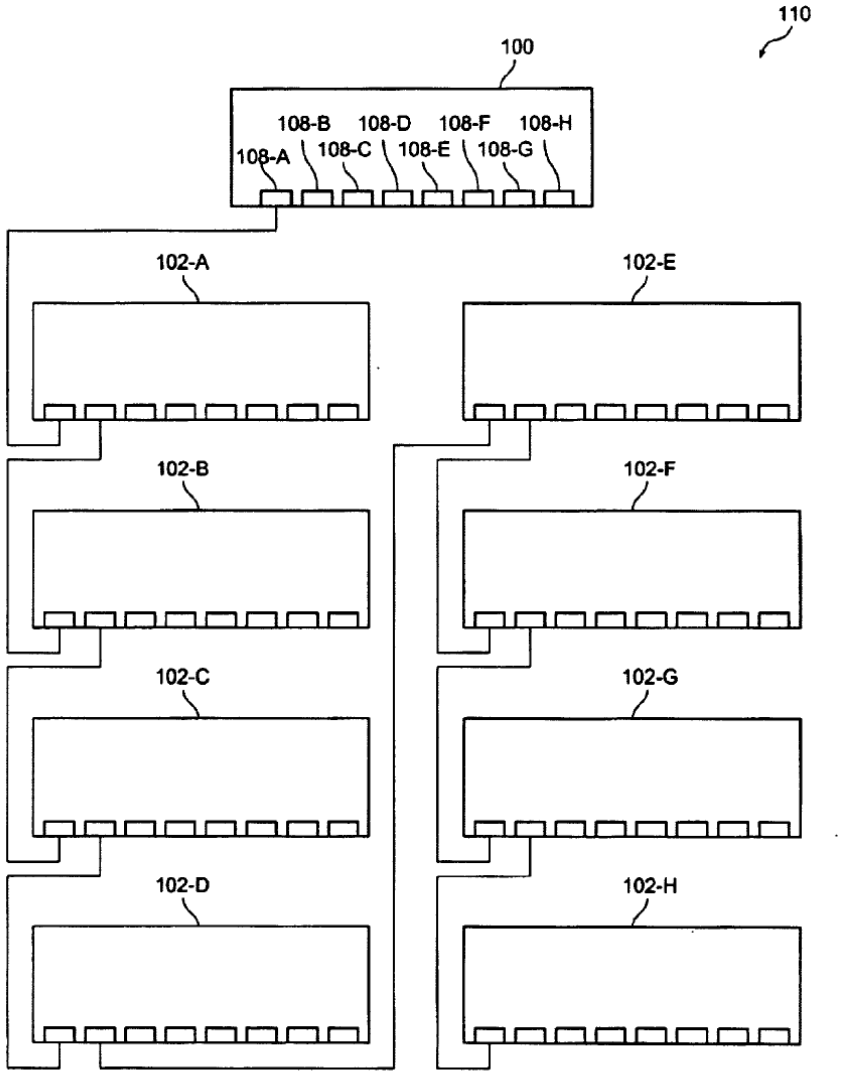
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

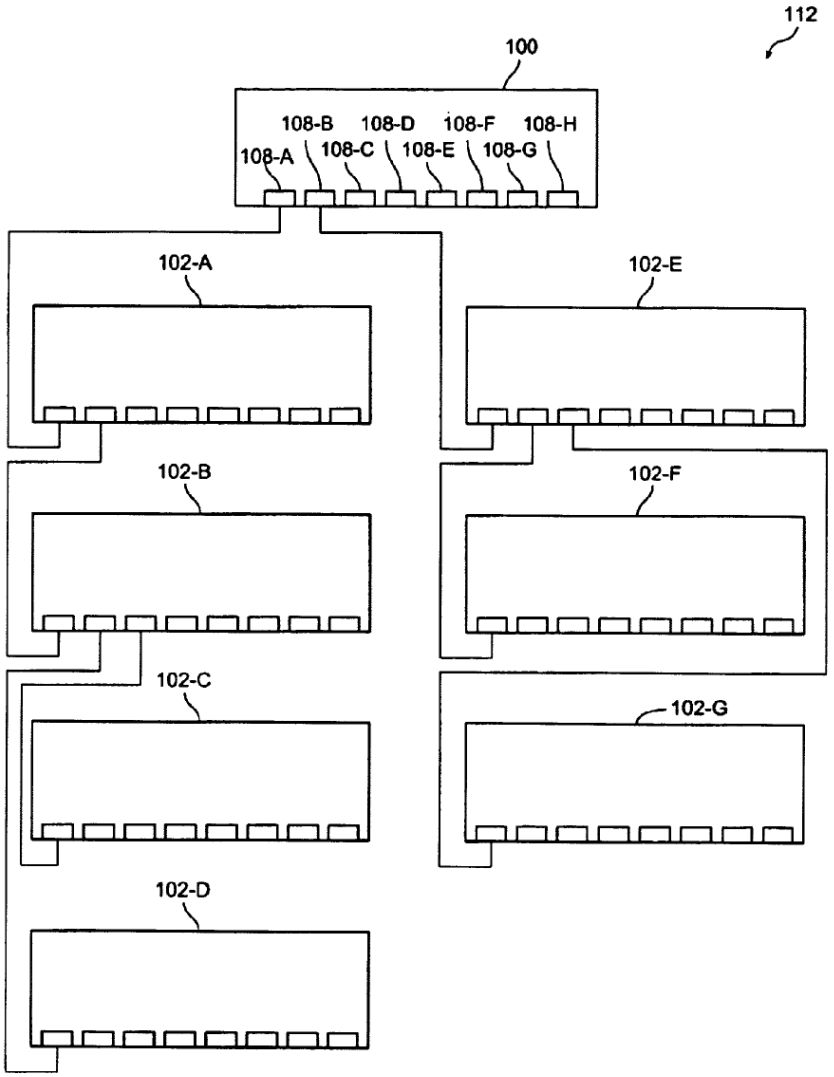
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

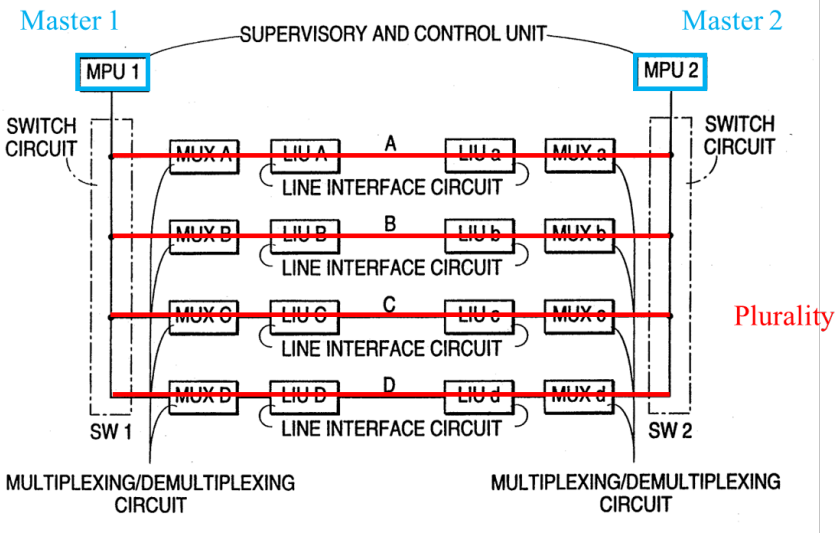
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

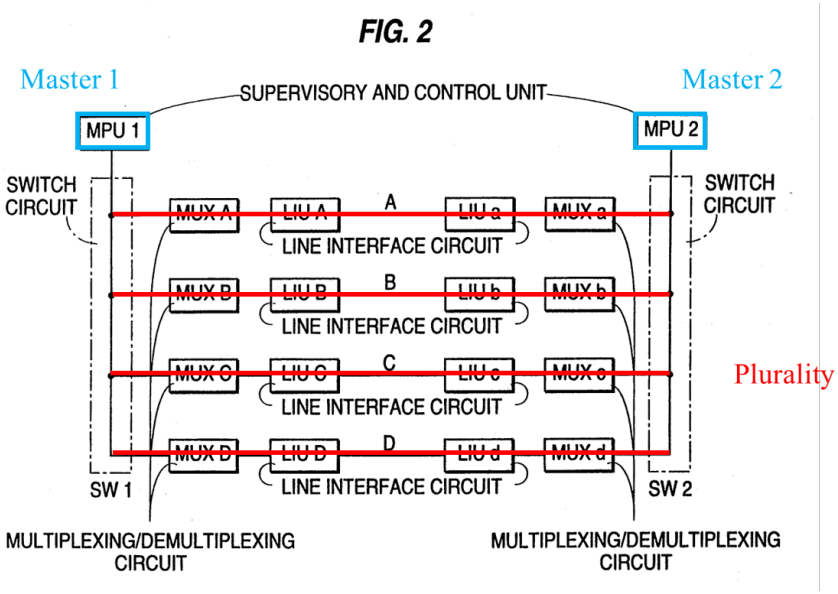
| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 11[c] | each slave unit comprising one or more ports to respective subscriber lines, and | The Reference discloses each slave unit comprising one or more ports to respective subscriber lines.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| 11[d] | a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween, | The Reference discloses a plurality of physical interface lines, which link the slave units in one or more daisy chains, in which the slave units are mutually connected in series by the physical interface lines therebetween.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling and Sugawara.<br><br>*See supra* at 1[d].<br><br>Cisco continues to make innovative contributions to the area of redundant stacked switch technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotated).<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |
| 11[e] | each daisy chain comprising at least a first slave unit connected by one of the physical interface lines to the first master unit and a last slave unit connected by another of the physical interface lines to the second master unit, | The Reference discloses each daisy chain comprising at least a first slave unit connected by one of the physical interface lines to the first master unit and a last slave unit connected by another of the physical interface lines to the second master unit.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Below are examples of such references. |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | **Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided.").<br>  <br><br>FIG. 2 (annotation added)<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| | | not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

| No. | '904 Patent Claim 11 | The Reference |
|---|---|---|
| 11[f] | wherein each of the slave units comprises a switch fabric comprising one or more switches, which convey data packets to respective ports on the switch to which the packets are addressed; and | The Reference discloses wherein each of the slave units comprises a switch fabric comprising one or more switches, which convey data packets to respective ports on the switch to which the packets are addressed.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |
| 11[g] | a pre-switch, which receives the data packets from one of the physical interface lines connected to the slave unit and passes those of the data packets that are addressed to any of the ports on the slave unit to the switch fabric, while passing packets not addressed to any of the ports on the slave unit for output through another of the physical interface lines. | The Reference discloses a pre-switch, which receives the data packets from one of the physical interface lines connected to the slave unit and passes those of the data packets that are addressed to any of the ports on the slave unit to the switch fabric, while passing packets not addressed to any of the ports on the slave unit for output through another of the physical interface lines.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 12 | The Reference |
|---|---|---|
| 12 | Apparatus according to claim 11, wherein each of the slave units is coupled to receive packets transferred thereto from the first and second master units over first and second ones of the physical interface lines, respectively, and wherein the pre-switch passes the packets received through the first and second physical interface line and addressed to any of the ports on the slave unit to respective first and second addresses in the switch fabric. | The Reference discloses apparatus according to claim 11, wherein each of the slave units is coupled to receive packets transferred thereto from the first and second master units over first and second ones of the physical interface lines, respectively, and wherein the pre-switch passes the packets received through the first and second physical interface line and addressed to any of the ports on the slave unit to respective first and second addresses in the switch fabric.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See also supra* at 11[f]. |

| No. | '904 Patent Claim 13 | The Reference |
|---|---|---|
| 13 | Apparatus according to claim 12, wherein in response to a reversal of a direction of data flow in the daisy chain, the first and second addresses are swapped in the pre-switch, so that substantially no reconfiguration of the switch fabric is required. | The Reference discloses apparatus according to claim 12, wherein in response to a reversal of a direction of data flow in the daisy chain, the first and second addresses are swapped in the pre-switch, so that substantially no reconfiguration of the switch fabric is required.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| 14[preamble] | In a network access multiplexing system, | The Reference discloses in a network access multiplexing system.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper. |
| 14[a] | in which a master unit is connected by a physical interface to a packet switched network, | The Reference discloses in which a master unit is connected by a physical interface to a packet switched network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL |

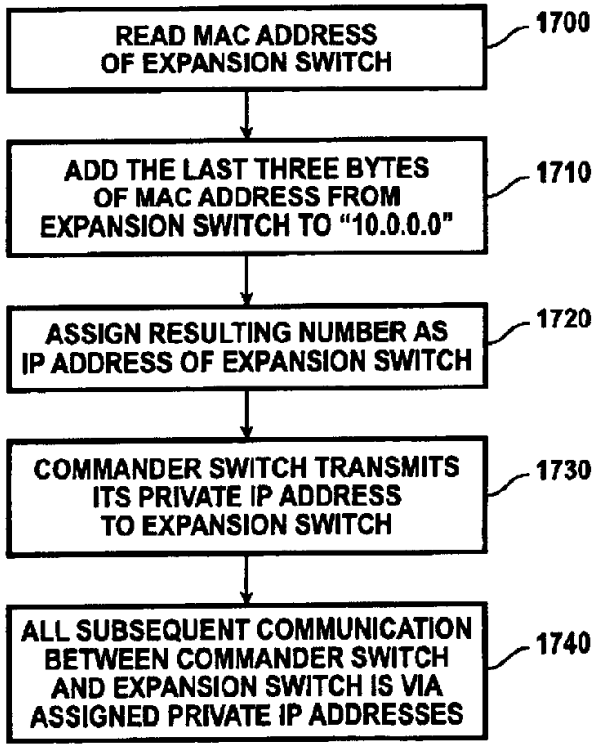
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper.<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626<br>• Duuvury '820<br><br>**Cisco Catalyst Press Release discloses**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2.<br><br>"With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3. |

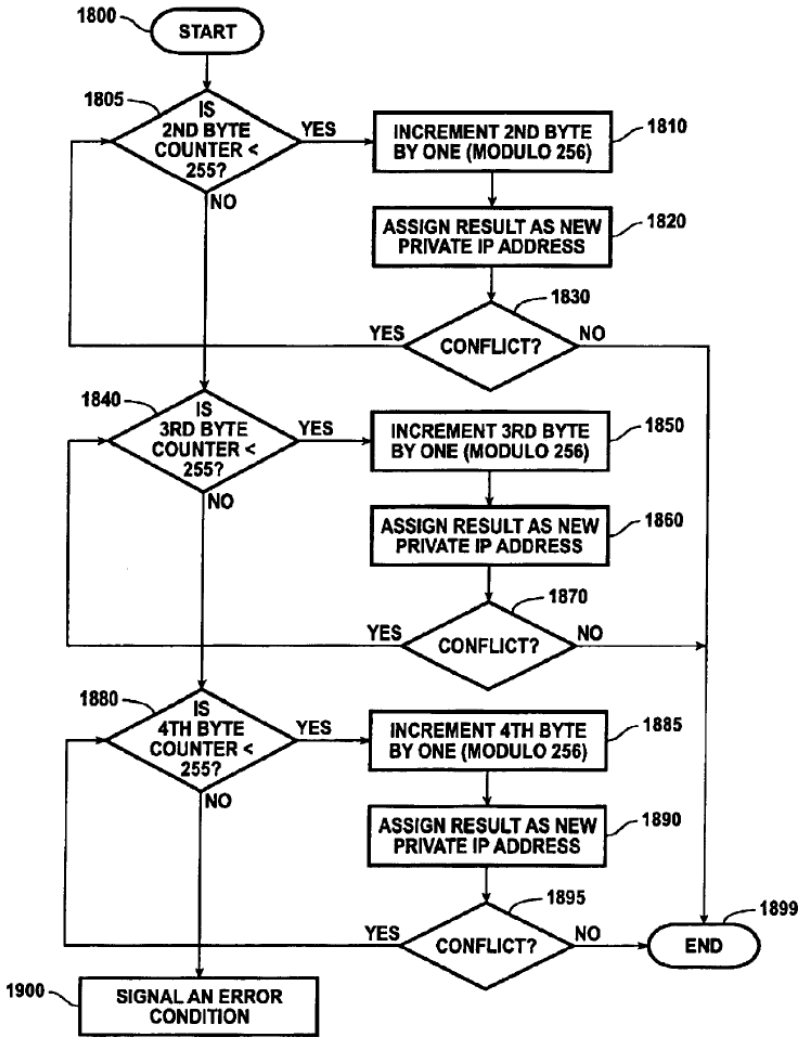
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | **Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract.<br><br>"If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31.<br>  <br><br>*FIG. 17*<br>Duvvury '626, FIG. 17. |

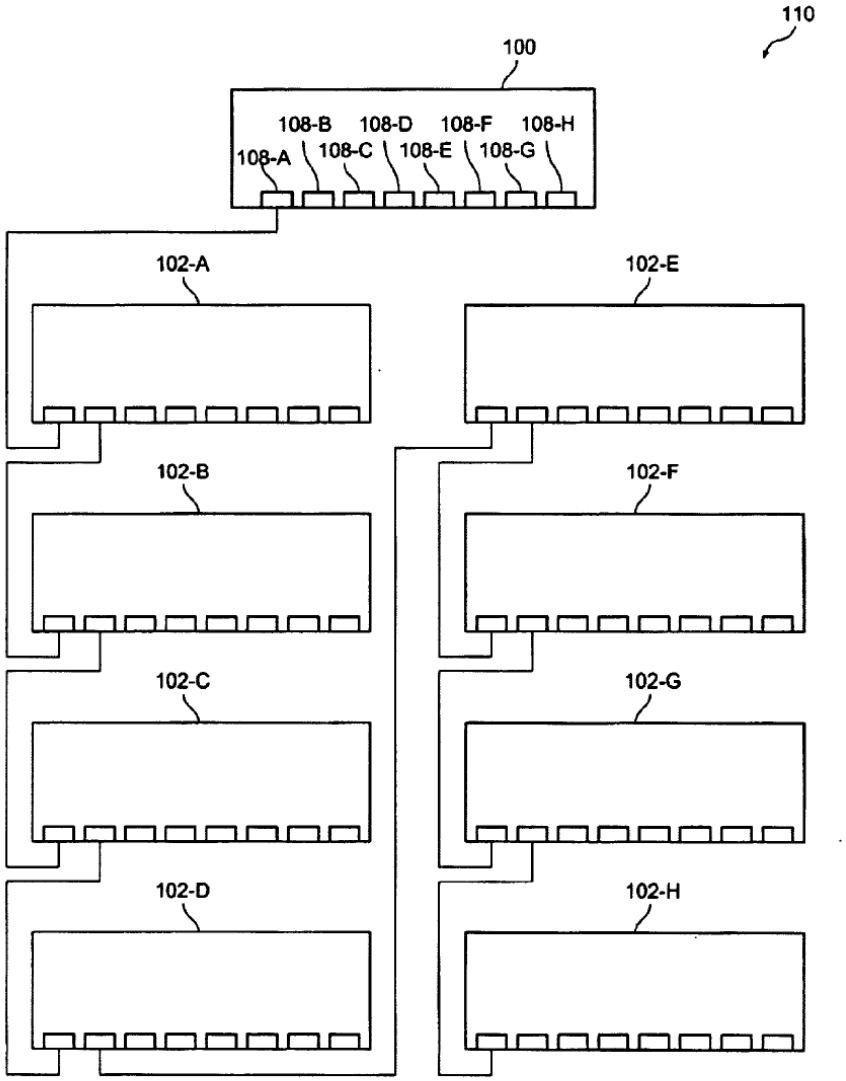
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

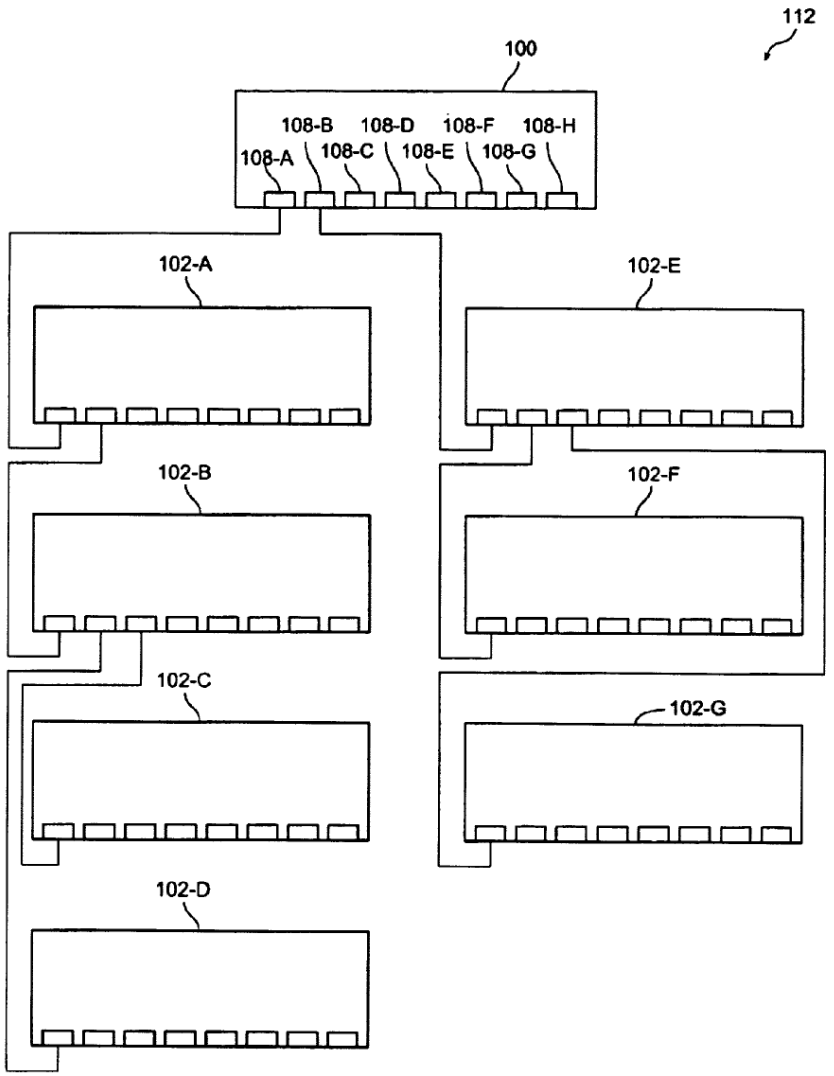
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10.<br><br>"Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

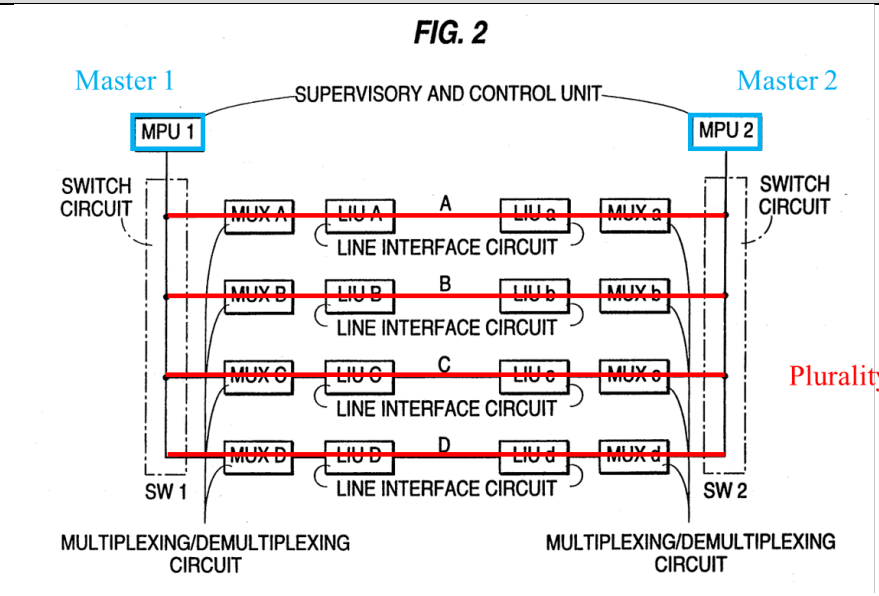
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 14[b] | slave unit configured to be coupled to the master unit in a daisy chain of such slave units, the slave unit comprising: | The Reference discloses slave unit configured to be coupled to the master unit in a daisy chain of such slave units, the slave unit comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper.<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433 |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | • Duvvury '626<br>• Duuvury '820<br><br>Below are examples of such references.<br><br>**<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2.<br><br>"With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

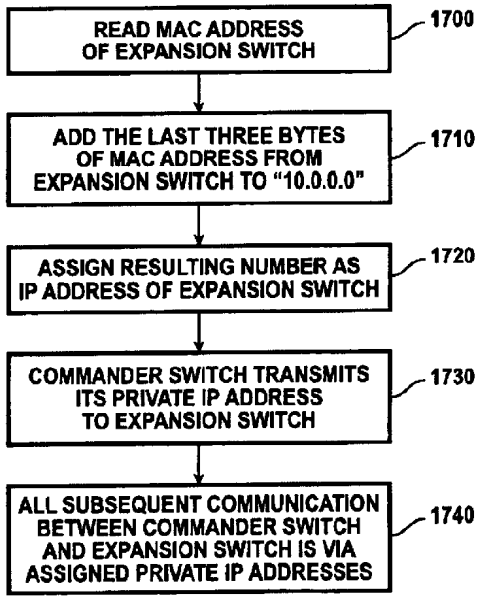
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

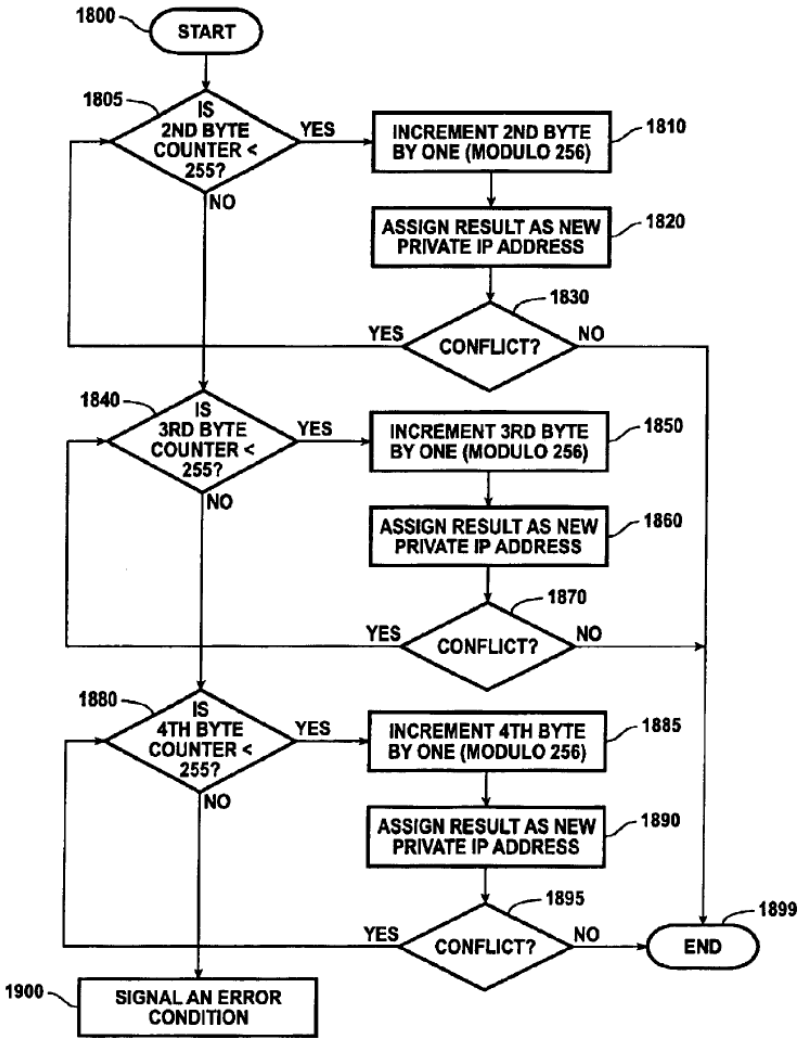
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 2 (annotation added)<br><br>**<u>Sugawara discloses</u>**:<br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.")<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract.<br><br>"If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

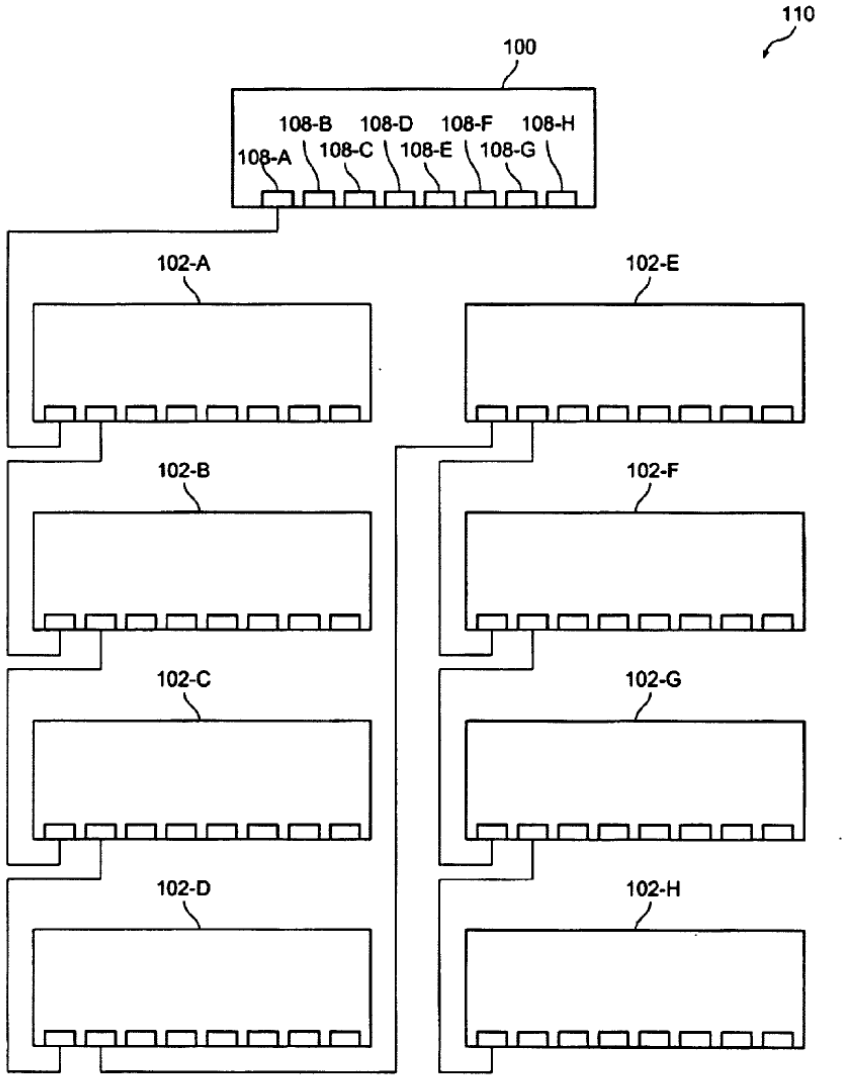
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

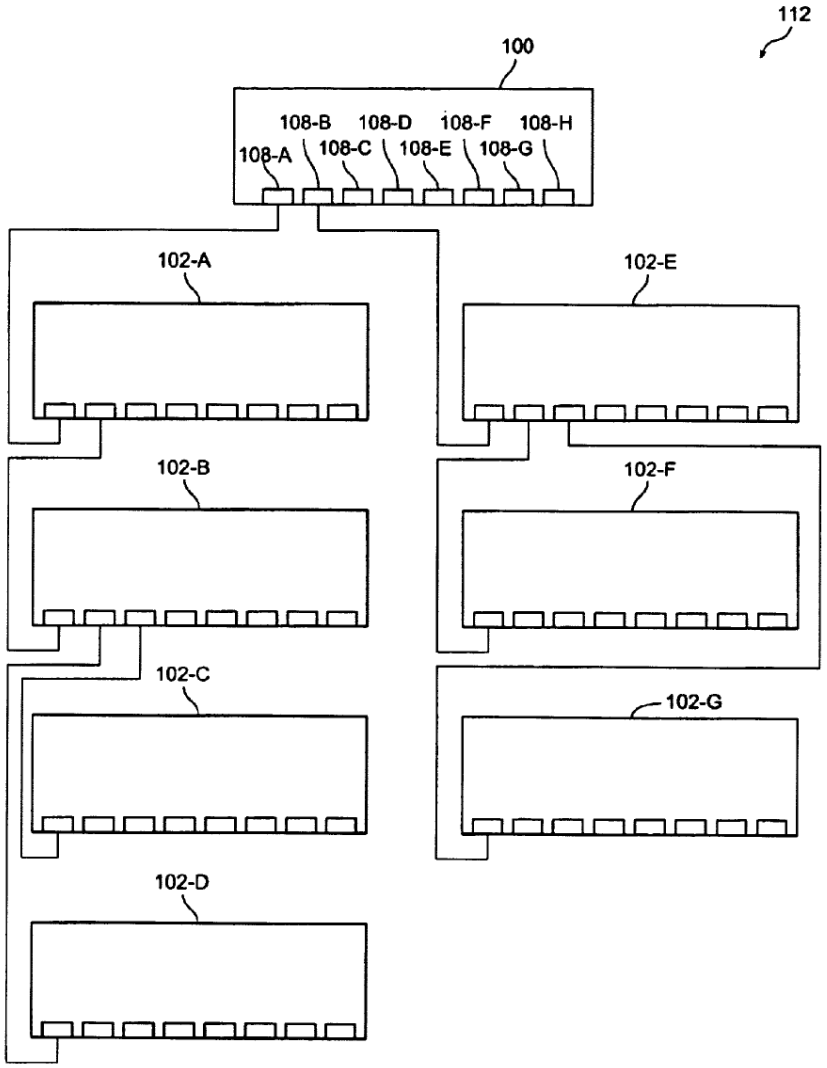
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

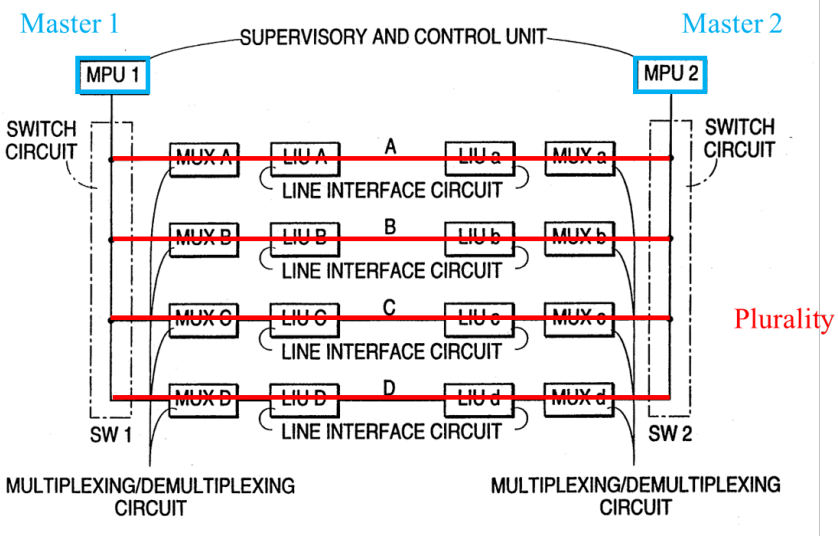
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

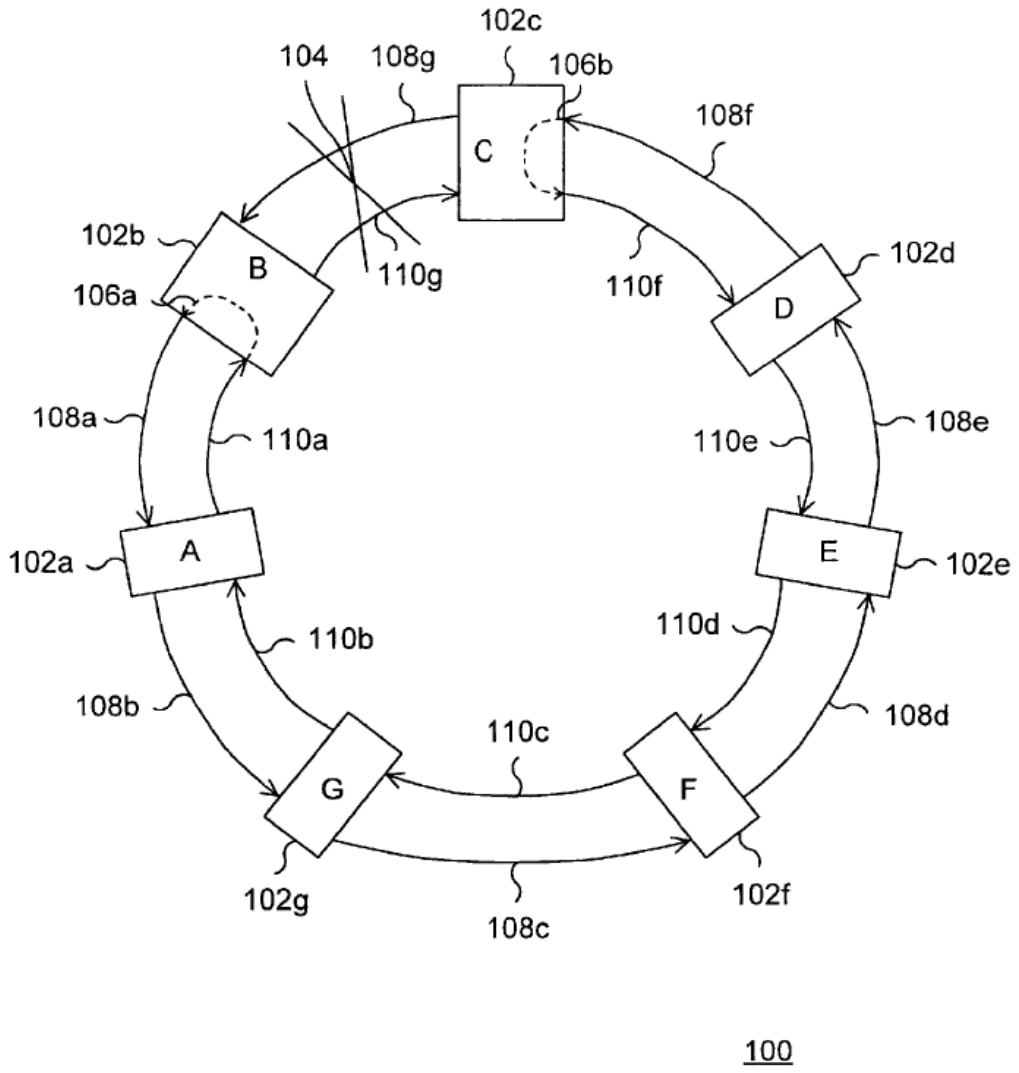
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 14[c] | a plurality of ports, for coupling the slave unit to respective subscriber lines; | The Reference discloses a plurality of ports, for coupling the slave unit to respective subscriber lines.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper. |
| 14[d] | first and second physical interfaces, coupled to exchange packets with preceding and succeeding units, respectively, along the daisy chain; | The Reference discloses first and second physical interfaces, coupled to exchange packets with preceding and succeeding units, respectively, along the daisy chain.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

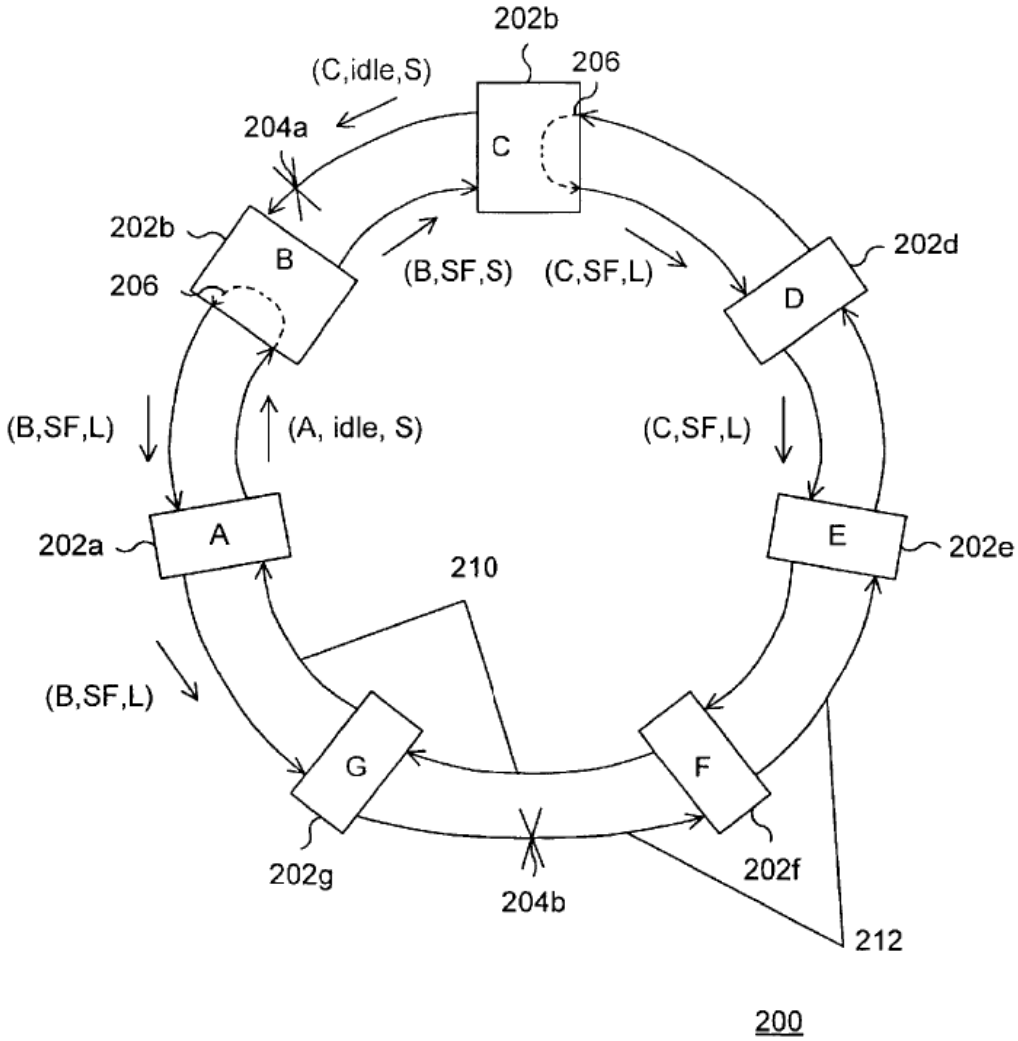
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper.<br><br>Cisco continues to make innovative contributions to the area of redundant stacked switch technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

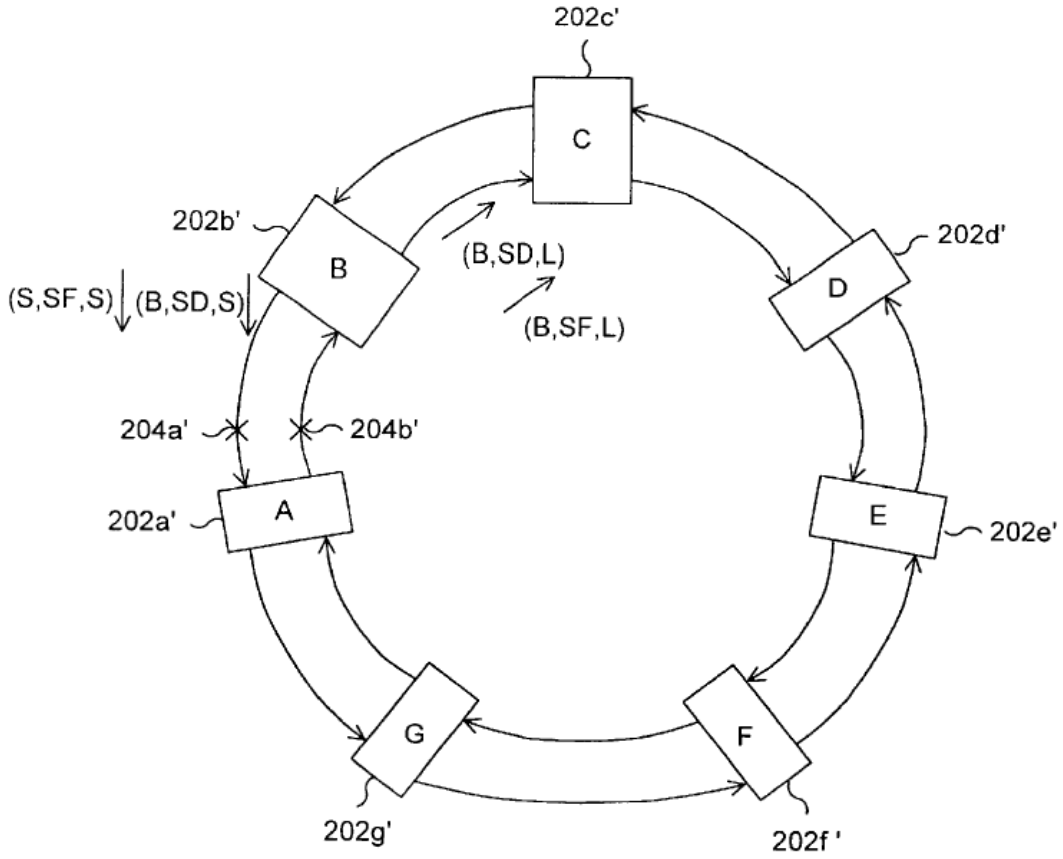
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotated).<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted."). <br><br> Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |
| 14[e] | a pre-switch, coupled to receive packets from the first physical interface and responsive to address data carried by the packets, to sort the packets such that packets addressed to the slave unit are retained, and packets addressed to the succeeding units are passed to the second physical interface, and | The Reference discloses a pre-switch, coupled to receive packets from the first physical interface and responsive to address data carried by the packets, to sort the packets such that packets addressed to the slave unit are retained, and packets addressed to the succeeding units are passed to the second physical interface. <br><br> To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper. <br><br> Cisco commercialized and patented technology relating to monitoring, detecting, and resolving faults without requiring a network reconfiguration *before* Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include: |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | • Daruwalla<br>• Nederveen<br>• Slater '421<br>• Petersen<br><br>**<u>Daruwalla discloses</u>**:<br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. The present invention provides a protection protocol which simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol will appropriately remove a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, would remove protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, Abstract. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 1. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 2. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 11.<br><br>"The present invention relates to computer networks. In particular, the present invention relates to a system and method for providing a protection protocol for fault recovery for a two line bi-directional ring network." Daruwalla, 1:8-11. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 1 shows an example of a two line bi-directional ring network. The ring network 100 is shown to include nodes 102 a-102 g. Each node is typically a computer with embedded processors and at least one network connection. Each node 102 a-102 g is shown to be bidirectionally coupled to two neighboring nodes 102 a-102 g via an inner connection ring 110 a-110 g and an outer connection ring 108 a-108 g. For instance, node 102 a is bidirectionally coupled to nodes 102 b and 102 g. The example of FIG. 1 also shows a problem 104 in the connection between node 102 b and node 102 c. When a problem is detected (such as a bi-directional line cut), the connection between nodes 102 b and 102 d wraps back upon itself, as shown by wraps 106 a and 106 b. In this manner, the connection problem 104 can be avoided." Daruwalla, 1:17-30.<br><br>"In a conventional SONET network, each message sent by a sending node to a receiving node typically needs the identification and location of the receiving node to arrive at the proper destination. Accordingly, manual configuration is typically needed in each node to store the identity and location of each other node in the ring network in order to provide for communication between the nodes in the network." Daruwalla, 1:31-44.<br><br>"In summary, for the protection mechanism to operate, each node needs to know the current ring map (current ring topology). What is needed is a system and method for providing fault recovery for two line bi-directional ring network that minimizes the need to keep track of other nodes in the ring network. Preferably, the system would not require reconfiguration of an internal map of the network when a new node is added to, or existing nodes are removed from the network. The present invention addresses such a need." Daruwalla, 2:23-31.<br><br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. According to the embodiment, when the problem is identified, the node sends a message identifying the problem to a second neighbor which is located at least one node away from the problem. The second neighbor then forwards the message to a third neighbor, unless the second neighbor is dealing with a situation that is higher in a hierarchy of situations than the problem described |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | in the message by the original node. In general, if the second neighbor's situation has a higher priority than the situation described by the original node, then the message is ignored and not forwarded. If, however, the message sent by the original node describes a situation with a higher priority than the situation being dealt with by the second neighbor, then, in general, the second neighbor's situation is ignored, at least for the moment, and the original node's message is forwarded to the next neighbor. In general, a higher priority request preempts a lower priority request within the ring. Exceptions are noted as rules of the protection protocol." Daruwalla, 2:35-58.<br><br>"The present invention provides a protection protocol that simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol automatically appropriately removes a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, removes protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, 2:59-3:3.<br><br>"A method according to an embodiment of the present invention for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:4-12.<br><br>"In another aspect of an embodiment of the present invention, a method for adding a new node to a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node prior to an addition of the new node; initiating a fault recovery procedure when the second node receives the first message; receiving a second |

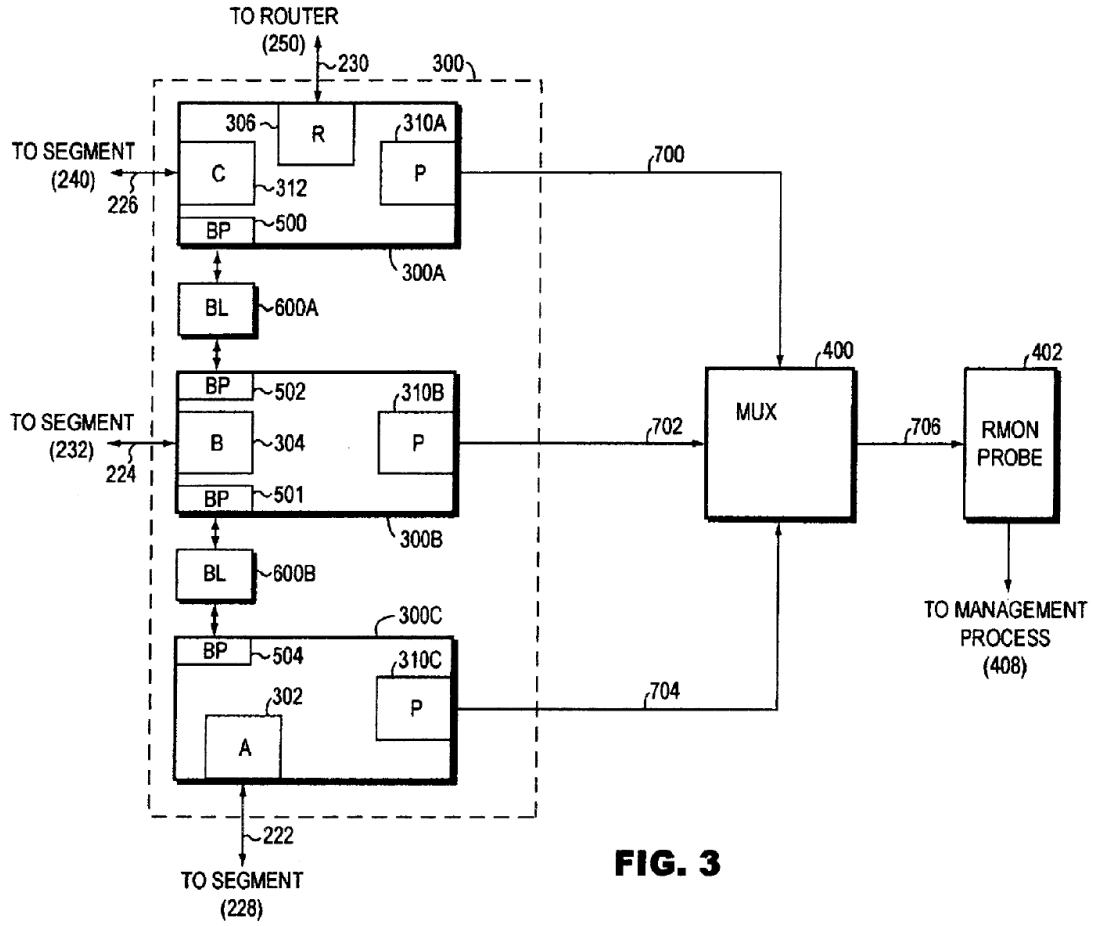
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | message from the new node; and entering an idle state when the second message is received." Daruwalla, 3:13-24.<br><br>"In yet another aspect of an embodiment of the present invention, a system for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The system comprises means for detecting a situation by a first node, wherein the first node is one of the plurality of nodes; means for sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and means for initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:25-35.<br><br>"FIG. 2 is block diagram of a ring network utilizing a protection protocol according to an embodiment of the present invention." Daruwalla, 3:40-42.<br><br>"FIGS. 4-6 are flow diagrams illustrating various rules within the protection protocol according to an embodiment of the present invention." Daruwalla, 3:46-48.<br><br>"FIGS. 8-12 are flow diagrams and a system diagram illustrating further rules of the protection protocol according to an embodiment of the present invention." Daruwalla, 3:52-54.<br><br>"FIG. 2 is a block diagram showing a ring network system utilizing a method of fault recovery according to an embodiment of the present invention. The ring network 200 is shown to include nodes 202 a-202 g. The nodes 202 a-202 g are shown to be coupled via an inner ring 210 in which the data flows in one direction, such as a clockwise direction. Additionally, the nodes 202 a-202 g are also shown to be coupled by an outer ring 212 in which data can flow in the opposite direction to the inner ring 210, such as in a counter-clockwise direction. The ring network 200 is shown to have a situation 204 a that requires protection, such as a ring wrap 206." Daruwalla, 5:35-45. |

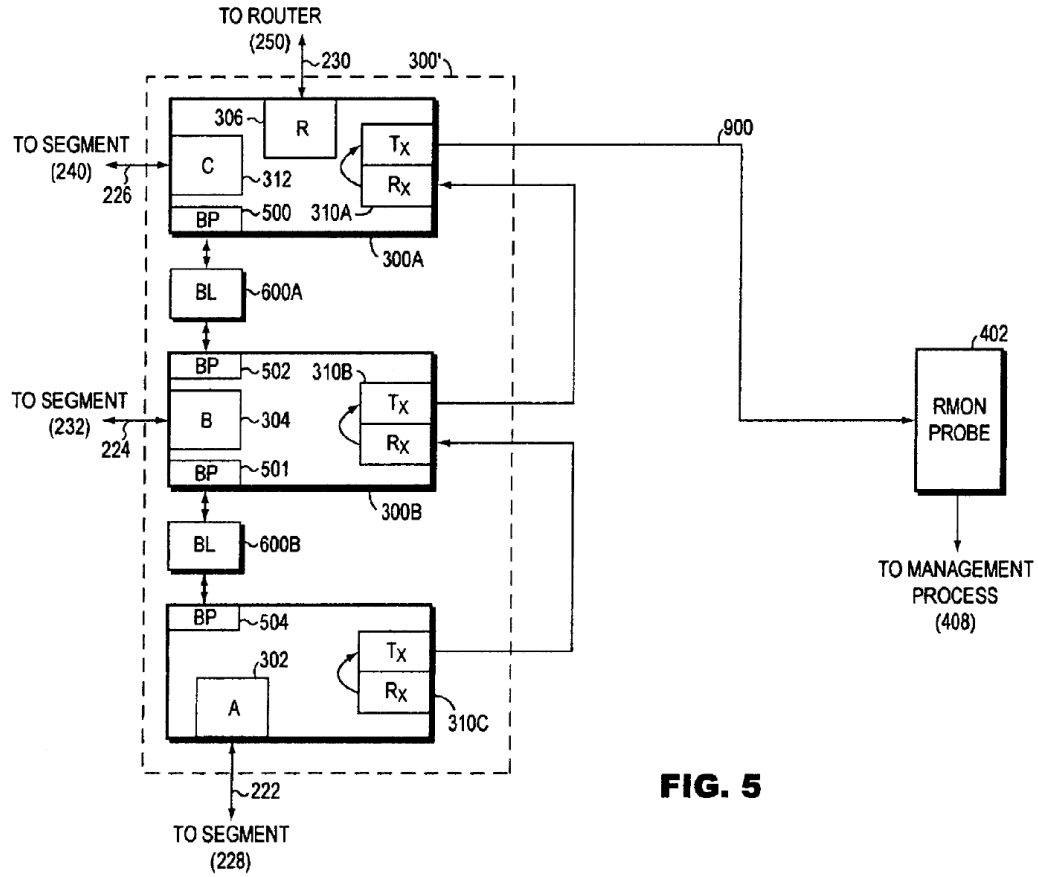
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 4 is a flow diagram of an example of a method according to an embodiment of the present invention implied by Rules 1-22. An APS packet is received via step 400. It is determined whether the APS packet has been sent along a long path via step 402. If the packet was not sent via a long path, then the APS packet is not forwarded via step 406. Accordingly, if the APS packet was sent via the short path, then the packet is not forwarded via step 406. If, however, the packet was sent through the long path via step 402, then the APS packet may be forwarded via step 404. Note that for this example of Rule (1), it is assumed that the long path message does not have to pass through a wrapped connection in order to be forwarded. Otherwise, if the long path message needs to pass through a wrapped connection in order to be forwarded, then the message will simply not be forwarded." Daruwalla, 6:21-36.<br><br>"FIG. 6 is a flow diagram illustrating Rules 4 and 5. A node detects a problem between the node and a first neighbor via step 600. The node performs a wrap away from the side on which the problem exists via step 602. A short path message is then sent to the first neighbor informing it of the problem via step 604. Additionally, a long path message is also sent to a second neighbor informing the second neighbor of the problem via step 604. The first neighbor then performs a wrap away from the side of the problem via step 606. The first neighbor also sends an IDLE message, indicating a wrapped status, on a short path to the node that detected the problem via step 608. This message is sent across the failed span. Note that IDLE messages do not get wrapped and are sent across failed spans if possible. Additionally, the first neighbor also sends a message on a long path toward the side without the problem via step 608." Daruwalla, 6:64-11.<br><br>"An example of the method described in FIG. 6 can be seen in FIG. 2. Node 202 b has detected a problem 204 a and performs a wrap 206 on the side on which the problem exists. Node 202 b also sends a short path message to the neighbor on the other side of the problem 204 a, which is node 202 c. Node 202 b also sends a long path message to its other neighbor node 202 a informing it of the problem. Node 202 c performs a wrap 206 on the side of the problem and sends an IDLE message on a short path to node 202 b. Node 202 c also sent a message on a long path toward the side without the problem to its neighbor 202 d." Daruwalla, 7:12-21. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 7 lists the hierarchy of priorities of Rule (8). For ease of reference, the hierarchy is separated into Class I-III. Class I is the highest priority, while Class III is the lowest priority. An example of a highest priority message in Class I is lockout. Lockout is an order stating that the ring network is not to wrap under any circumstances." Daruwalla, 7:22-26.<br><br>"Examples of the next priority listed in Class II are forced switch and signal fail. Forced switch indicates that the ring network is configured to wrap at the point of the forced switch. Signal fail is a situation where either two nodes cannot communicate with each other, or one node cannot hear the other node. An example of a signal fail is a physical break in the communication lines between two nodes." Daruwalla, 7:27-33.<br><br>"Note that members of Class II can co-exist (Rule 9). For example, multiple forced switches and signal fails can co-exist. Additionally, members of Class I can co-exist (Rule 10). For example, multiple lockouts in a single ring network can co-exist. However, situations in Class III cannot co-exist with other situations (Rule 11). For example, a signal degrade cannot co-exist with a wait-to-restore." Daruwalla, 7:52-58.<br><br>"When there are multiple requests of the same priority within Class III, the first request to complete a long path signaling will take priority (Rule 13). For example, if there are two signal degrades located on the same ring network, then the first signal degrade which completes the long path signaling will take priority over the other signal degrade. By not allowing members of Class III to co-exist with one another, partitioning of the ring network is avoided." Daruwalla, 7:59-65.<br><br>"In case of two equal requests within Class III on both inner and outer rings of the ring network, the tie is broken by choosing a request on one of the rings, such as the outer ring request (Rule 14). For example, if a signal degrade occurs both on the inner and outer rings, then a request on a predetermined ring, such as the outer ring, will take priority over the other requests." Daruwalla, 7:66-8:5. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 8 is a flow diagram illustrating Rules (9), (10), (11), (13), and (15). Note that the flow diagram described in FIG. 8 is merely an example of one way in which the rules of the method according to the embodiment of the present invention can be executed. For example, the determination of whether the long path message is a Class I request via step 802 or a Class II request via step 810 can be in reverse order." Daruwalla, 8:6-11.<br><br>"A wrapped node receives a long path message via step 800. It is then determined if the long path message is a Class I request via step 802. The classes used in FIG. 8 are meant to correspond with the example of classes defined in FIG. 7. If the long path message is a Class I request, then it is determined if a local situation also has a Class I request via step 804. A local situation includes scenarios such as when a node detects a situation or problem, or when a node is made aware of a problem or situation via a short path message from its neighbor. If a local situation is not a Class I request via step 804, then any existing local wraps are unwrapped and the long path message is forwarded via step 806. If, however, a local situation is a Class I request via step 804, then the connections are already unwrapped or was never wrapped, and the long path message is forwarded via step 808." Daruwalla, 8:12-26.<br><br>"FIG. 12 is a flow diagram illustrating rules (20) and (21) of the method according to the embodiment of the present invention. A wrapped node determines that a problem has been cleared via step 1200. It then enters a wait-to-restore state via step 1202. It is then determined if its neighbor is the same neighbor as previously noted via step 1204. The node can save the source of a short path message at the time of wrap initiation to note the identity of its neighbor. If the current neighbor is not the same as the previous neighbor via step 1204, then an IDLE state is entered via step 1206. If, however, the current neighbor is the same as the previous neighbor via step 1204, then it is determined whether a pre-determined wait-to-restore time has expired via step 1208. Once the pre-determined wait-to-restore time has expired, then the node enters an IDLE state via step 1206." Daruwalla, 12:60-13:6.<br><br>"A method and system for fault recovery for a two line bi-directional network has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor." Daruwalla, 13:7-19. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | **Nederveen discloses**:<br>"A technique for use in gathering network activity-related information from cascaded network switches is provided. Using this technique, the information can be gathered without substantially reducing performance of the cascaded switches. In one embodiment, a single remote monitoring probe is connected via respective connections to each of the switches so as to receive the information from the switches. In another embodiment, only one of the switches is connected to the probe, and the other switches transmit their respective portions of the information to the switch connected to probe. The switch connected to the probe provides these portions of the information, as well as, any of its respective activity-related information to the probe. In this latter embodiment, the switches may be connected by dedicated connections and switch ports that are used solely for communicating the activity-related information." Nederveen, Abstract. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>**FIG. 3**<br>Nederveen, FIG. 3. |

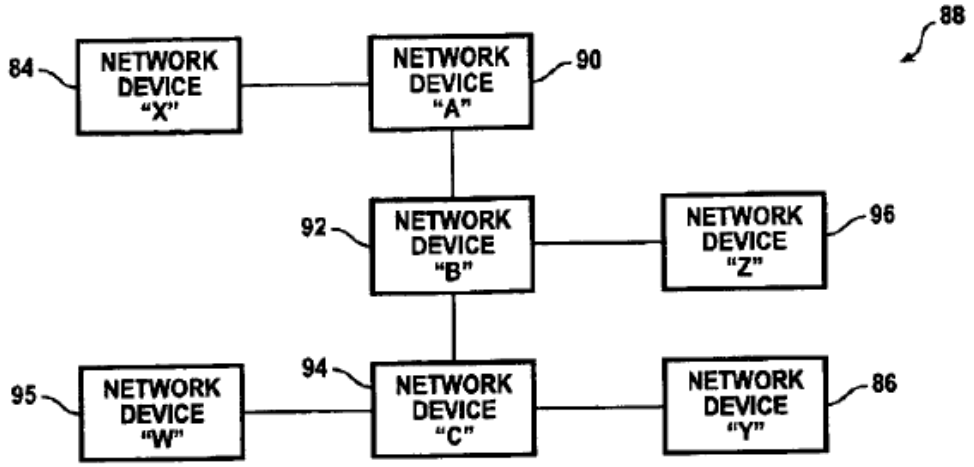
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>**FIG. 5**<br><br>Nederveen, FIG. 5.<br><br>"Thus, it would be desirable to provide a stacked switch monitoring technique that permits efficient offloading of raw data processing from the stacked switches, requires only a minimal number of specialized network entities to gather and process such raw data, and does not result in substantial degradation of stacked switch performance." Nederveen, 4:38-43. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Accordingly, the present invention provides a technique for remote monitoring of a switch network that overcomes the aforesaid and other disadvantages and drawbacks of the prior art. More specifically, in one aspect of the present invention, a technique is provided for gathering information that may be useful in network management (e.g., switch port activity-related information), from switches in the network that are in a stacked configuration. The information is gathered from the stacked switches by a single network entity (e.g., an SNMP remote monitoring probe) in such a way that it does not substantially degrade the performance of the switches. This is accomplished, in one embodiment of the technique of the present invention, by connecting the switches via respective connections to a multiplexer that selectively connects the switches, according to an arbitration scheme, to the single network entity. The entity gathers respective portions of the information from switches when it is connected to the switches by the multiplexer. The information gathered by the entity may be provided to another network entity (e.g., an SNMP management node) in order to permit the other entity to use that information in managing the network." Nederveen, 4:46-67.<br><br>"In another embodiment of the technique of the present invention, only one of the switches is connected to the single information gathering entity. The switches that are not connected to the entity transmit, via respective dedicated ports and connections (i.e., ports and connections that are used solely for network information gathering activities), their respective portions of the information to the switch that is connected to the entity. The switch that is connected to the entity transmits, via a respective dedicated port and connection, the information received from the other switches, as well as, its own information to the entity." Nederveen, 5:1-11.<br><br>"FIG. 3 is a schematic, functional block diagram illustrating in greater detail the construction of the stacked switch network shown in FIG. 2." Nederveen, 5:26-28.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network configured to employ another embodiment of the present invention." Nederveen, 5:32-34. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIGS. 2-5 illustrate features of a computer network 200 wherein embodiments of the present invention may be advantageously practiced. Network 200 comprises a stacked switch network 300 which interconnects a plurality of network segments 228, 232, 240, and 251. Each segment 228, 232, 240 comprises one or more local area networks having computer endstations (not shown). Segment 251 is a network router segment that comprises network router 250. Each segment 228, 232, 240 is coupled via a respective communications link 222, 224, 226 to a respective port 302 (i.e., port A), 304 (i.e., port B), 312 (i.e., port C) of the switch network 300. Likewise, the router 250 of router segment 251 is coupled via a respective trunk line 230 to router port 306 (i.e., port R)." Nederveen, 5:46-59.<br><br>"Stacked switch network 300 comprises a plurality of data network switches 300A, 300B, 300C (e.g., Catalyst 3900™ series switches of the type commercially available from the Assignee of the subject application) coupled together via conventional stack link bus connection logic 600A, 600B. More specifically, logic 600A couples a stack link bus port and associated logic 500 in switch 300A to a stack link bus link port and associated logic 502 in switch 300B. Similarly, logic 600B couples another stack link bus port and associated logic 501 in switch 300B to a stack link bus port and associated logic 504 in switch 300C. It should be understood that although, as is shown in FIG. 3, switches 300A and 300B, and switches 300B and 300C, may be coupled serially together by separate respective logic elements 600A, 600B, each of the switches 300A, 300B, 300C may be coupled together via a single respective stack link bus port in the switch to a single stack link bus connection logic block (not shown, e.g., of the type that is commercially available under the tradename Catalyst Matrix™ from the Assignee of the subject application). Further alternatively, depending upon the particular design and functionality of the ports 500, 501, 502, and 504, and the control and forwarding logic (whose operation will be described more fully below) in the switches 300A, 300B, 300C, the circuitry in logic 600A, 600B may instead be comprised in the ports 500, 501, 502, and 504 and/or control and forwarding logic, and therefore, in this alternative configuration, the logic 600A, 600B in the network 300 may be replaced by simple connection means (e.g., cable connectors)." Nederveen, 6:29-57. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Each switch 300A, 300B, 300C includes a respective internal bus (e.g., element 800 in switch 300C) that is coupled via at least one stack link bus port and associated interface logic (e.g., 504 in switch 300C) to external stack link bus connection logic (e.g., element 600B in switch 300C). Each switch 300A, 300B, 300C also includes respective programmable control and forwarding logic (e.g., element 802 in switch 300C) comprising processing, memory, and other circuitry for storing and learning configuration information (e.g., source and destination MAC addresses of messages received by the switch, switch bridging table, switch segments' spanning tree and virtual local area network information, etc.), and for providing appropriate commands to other elements (e.g., the switch ports) to cause data messages received by the switch to be forwarded to appropriate network segments coupled to the switch based upon this configuration information. In each switch, the switch's port logic circuitry (e.g., port A logic 302 and port P logic 310C in switch 300C) and control and forwarding logic are coupled to each other via that switch's respective internal bus. The stack link bus port and associated logic in each switch 300A, 300B, 300C may comprise a Catalyst™ stack port line interface card (commercially available from the Assignee of the subject application) inserted into a bus expansion slot (not shown) in the switch. Although not shown in the Figures for purposes of clarity of illustration, each switch 300A, 300B, 300C in network 300 typically will include tens or hundreds of ports coupled to network segments." Nederveen, 6:58-7:19.<br><br>"The control and forwarding logic and stack link bus port and associated logic in each switch, and the logic 600A, 600B, are configured to together implement conventional techniques for permitting the switches 300A, 300B, 300C to function together as a single logical/virtual switch. More specifically, when configured in the stacked arrangement 300, after the switches 300A, 300B, 300C and logic 600A, 600B are initially activated, they execute initial power-on self-diagnostics, and thereafter, enter a "stack discovery" mode of operation." Nederveen, 7:20-29.<br><br>"In the stack discovery mode of operation, the control and forwarding logic in each switch 300A, 300B, 300C first "senses" that its switch is coupled to logic 600A and/or 600B, and then determines the particular configuration of the stacked switch network 300, using suitable conventional autosensing/autoconfiguration techniques. The control and 261orwarding logic in the switches 300A, 300B, 300C then assigns to the switches respective unique |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | identification numbers (e.g., based upon unique identification numbers of respective ports of the logic 600A, 600B to which the switches are coupled)." Nederveen, 7:30-40.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network 300′ configured to employ another embodiment of the present invention. It should be understood that unless specifically stated to the contrary, the structure and operation of the network 300′ are substantially the same as the structure and operation of network 300. In network 300′, each of the dedicated ports 310A, 310B, 310C comprises a respective transmit portion and receive portion, referenced in FIG. 5 as RX and TX, respectively." Nederveen, 11:20-29.<br><br>**Slater '421 discloses**:<br>"A method and apparatus for discovering paths to other network devices includes a protocol and network management application that can be executed on network devices. The Ethernet protocol is used to detects paths to other network devices, knowing only the Ethernet address of the destination. A discovery protocol is extended to add hop probe and hop probe reply Type-Length-Value fields in a variable-length list. The hop probe fields contain a hop count, a destination Ethernet address, and a source Ethernet address. When a hop probe is received by a network device, the hop count field is decremented by one and the hop probe is forwarded. Packet received with a hop count of one are not forwarded and a hop probe reply is sent back to the Ethernet source address of the hop probe. The hop probe reply fields contain a destination Etherned address and a source Ethernet address." Slater '421, Abstract. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br><br>Slater '421, FIG. 6.<br><br>"Partly as a result of the increased complexity of networks, network administrators must often troubleshoot problems with their network. Two classes of network problems often faced by network administrators are "reachability" problems and performance slowdowns. Reachability problems occur when one or more network devices cannot be accessed through a network, and can be caused by hardware or software failures, cabling problems, or any of several other types of difficult-to-diagnose problems that can occur in a network." Slater '421, 7:11-20.<br><br>"Some of the world's largest networks today rely on the TCP/IP suite of networking protocols. With a relatively small kit of basic tools, network administrators can learn much about an internetwork. "Ping" and "traceroute" commands, "show" commands, and "debug" commands (all of which are typically available via the basic management interface on a network device) form the core of the network administrator's internetwork toolkit. Ping and |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | traceroute commands can be useful tools in determining where failures are occurring, but they are cumbersome to use, and require knowledge of the IP address or host name of the destination network device. The show commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data. Finally, debug commands provide clues about the status of a network device and other network devices directly or indirectly connected to it. Because debug commands can create performance slowdowns, they must be used with great care, and using the wrong debug command at the wrong time can exacerbate problems in already poorly performing networks." Slater '421, 7:55-8:8.<br><br>"Embodiments of the present invention as illustrated herein use the Cisco™ Discovery Protocol ("CDP") to automatically detect paths to specified network devices in Ethernet LANs. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task." Slater '421, 9:10-15.<br><br>"CDP is a media-independent device discovery protocol which can be used by a network administrator to view information about other network devices directly attached to a particular network device. In addition, network management applications can retrieve the device type and SNMP-agent address of neighboring network devices. This enables applications to send SNMP queries to neighboring devices. CDP thus allows network management applications to discover devices that are neighbors of already known devices, such as neighbors running lower-layer, transparent protocols." Slater '421, 9:16-26.<br><br>"It is to be understood that the present invention is not limited to devices that are compatible with CDP. CDP runs on all media that support the Subnetwork Access Protocol ("SNAP"), including LAN and Frame Relay. CDP runs over the data link layer only. Each network device sends periodic messages to a multicast address and listens to the periodic messages sent by others in order to learn about neighboring devices and determine when their interfaces to the media go up or down. Each device also advertises at least one address at which it can receive SNMP messages. CDP messages, or "advertisements," contain holdtime information, which indicates the period of time a receiving device should hold CDP information from a neighbor before discarding it. With CDP, network management applications can learn the device type |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | and the SNMP-agent address of neighboring devices. This process enables applications to send SNMP queries to neighboring devices." Slater '421, 9:27-43.<br><br>"It should be noted that-normally, CDP packets according to aspects of the present invention are transmitted at regular intervals (e.g. once every 60 seconds). However, in embodiments of the present invention, when a Hop Probe or Hop Probe Reply needs to be forwarded by a network device, the network device is commanded to send a CDP packet immediately." Slater '421, 16:66-17:5.<br><br>"The present invention is much faster than the previous method that involved logging in to each intermediate network device, entering the "show cdp neighbors" command, and interpreting the output to find the next hop along the path to the destination network device. Also, the present invention allows individual users, such as network administrators, to execute a tool to manually discover paths through a network of Ethernet switches. The present invention can be used by network management software to automatically map the topology of clusters of network devices, such as Ethernet switches. Finally, the present invention is useful in loop detection. Enhancements to Spanning Tree and other bridge-level routing protocols can test proposed changes to switch topology prior to making them." Slater '421, 17:6-20.<br><br>**Petersen discloses**:<br>"Methods and apparatus for enabling communication between a source network device and one or more destination network devices are disclosed. A system enabling communication between a source network device and one or more destination network devices includes a switch and a ring interconnect. The switch is adapted for connecting to the source network device and the one or more destination network devices. More particularly, the switch is capable of storing data provided by the source network device and retrieving the data for the one or more destination network devices. The ring interconnect is adapted for connecting the source network device and the one or more destination network devices to one another. In addition, the ring interconnect is capable of passing one or more free slot symbols along the ring interconnect. Thus, the ring interconnect is capable of expanding when one or more of the free slot symbols are each replaced by a frame notify message indicating that the data has |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | been stored by the switch for retrieval by the one or more destination network devices." Petersen, Abstract. |

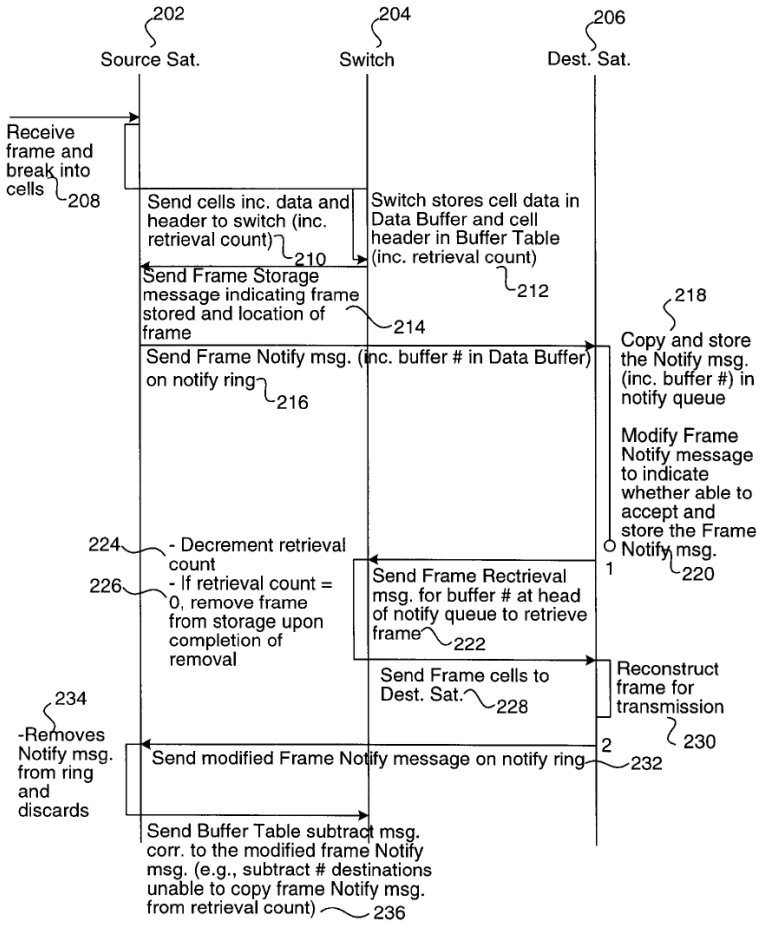



***FIG. 2***

Petersen, FIG. 2.

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The present invention relates to a mixed topology data switching system that combines a radial interconnect with a ring interconnect. More particularly, the radial interconnect permits devices to store and retrieve data using a switch, while the ring interconnect permits devices along the ring interconnect to provide notification that data has been stored for retrieval, as well as provide feedback regarding the ability or inability to retrieve such data." Petersen, 1:34-41.<br><br>"In controlling the flow of network traffic through a switching system, it is often desirable to provide feedback to the source of the data. For instance, although a transmitting device, hereinafter referred to as a "source device," may transmit or forward data to a receiving device, hereinafter referred to as a "destination device," the destination device may be incapable of handling the data. In these circumstances, the source device is often unaware that the data was not accepted by the destination device, complicating switch management. Common solutions to the problem of switch traffic management have included ensuring that all intended destination devices are "ready to receive" prior to transmitting data on a ring or bus interconnect, or insisting that each intended destination device send an explicit acknowledgement back to the source device. Both of these approaches result in reduced efficiency of the interconnect scheme. By way of example, in a ring network, such acknowledgment is typically provided in the data frame being transmitted. As another example, in other interconnect schemes, each such device may send a separate acknowledgment, therefore adding to the traffic on the network. Accordingly, it would be desirable if a traffic management scheme were established which could provide such feedback to the source of the data while minimizing traffic management overhead." Petersen, 2:8-32.<br><br>"According to one embodiment, the present invention combines the use of two data transport methods: a point-to-point radial interconnect and a ring interconnect. The radial interconnect connects interface devices to each other through the services of a central switch device to permit the transport of data. Typically, a single interface has a single dedicated radial interconnect to the central switch. These interface devices are further connected to one another via a ring interconnect to convey retrieval notifications regarding forwarding of the data (by source devices) and receipt of the data (by destination devices)." Petersen, 2:36-46. |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Each radial interconnect provides a narrow, high baud-rate connection to convey to the actual data from and to the associated interface without being burdened by the unrelated traffic for the remaining interfaces in the system. This is accomplished through the use of a central switch device, which stores and retrieves data for the various interfaces in the system. As will be apparent from the following description, this architecture provides numerous advantages over a wide parallel bus or ring." Petersen, 2:47-55.<br><br>"The ring interconnect may be used to convey a "retrieval notification"(i.e., retrieval message) that may be observed by all potential retrieving interfaces. The retrieval notification notifies specific devices ("destination devices") or interfaces that one or more frames addressed to them are available from the switch device. Moreover, the ring interconnect permits each destination device to provide feedback to the source device letting the source know whether the destination has accepted the notification provided by the source device and therefore whether the destination can retrieve the data intended for it. The feedback is particularly useful in buffer management applications. In this manner, an efficient and flexible data transport and retrieval notification system that includes a feedback path to the source of the data is provided." Petersen, 2:56-3:3.<br><br>"FIG. 2 is a process flow diagram illustrating a method of providing network communication according to an embodiment of the invention." Petersen, 3:9-11.<br><br>"FIG. 2 is a process flow diagram illustrating in further detail a method of providing network communication in the above-described system according to an embodiment of the invention. As shown, process steps performed by a source device 202 are illustrated along an associated vertical line, steps performed by a switch 204 are illustrated along another vertical line, and steps performed by a destination device 206 are illustrated along still another vertical line." Petersen, 4:50-57.<br><br>"When the frame is stored by the switch 418, the source device preferably receives an acknowledgment that the data has been stored. Thus, to provide this feedback, a frame storage message (i.e., storage reply) is sent from the switch 418 on the channel 416 to the channel interface 414. The frame storage message is then provided to the notify ring interface as |

| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | shown at 430 and sent on the notify ring. Once this acknowledgment is received by the interface device 402, the designated destination devices may be notified via notify ring interface 424. As described above, a Frame Notify message may be sent via the notify ring interface 424 to the destination devices. More particularly, the Frame Notify message may identify one or more destination devices for the frame and specify the location of the frame to be retrieved. By way of example, the location of the frame to be retrieved by the destination devices may be designated by a buffer number 430. In addition, the destination devices for the frame may be specified in the Frame Notify message through a notify queue map 426. More particularly, the notify queue map 426 may specify a notify queue associated with a particular destination device. The notify queue may be expressly designated through the use of one or more bits as well as implied through the specification of a priority level for the data. The notify queue map 426 will be described in further detail with reference to FIG. 13. The notify ring interface 424 then creates a Frame Notify message including the notify queue map 426 and the buffer number 430 which is then sent on an outbound interface of the notify ring 432." Petersen, 7:55-8:16.<br><br>"As described above, the notify ring may be expanded to accommodate communication between interface devices. The communication between the interface devices and the switch is therefore performed on one or more channels rather than the notify ring. As a result, the flexibility of the notify ring does not effect the speed with which the interface devices may communicate with the switch. Thus, where a single port operates at a faster speed than the channels, multiple channels may be grouped together. In this manner, the speed with which the switch may communicate with the interface devices may be maximized." Petersen, 20:27-36.<br><br>"The present invention provides a mixed topology data switching system that combines a point-to-point radial interconnect with a ring interconnect to maximize the speed of network traffic. The radial interconnect provides a narrow, high baud-rate connection to convey the data traffic for just the interface in question, without being burdened by all of the unrelated traffic for the remaining interfaces in the system. At the same time, the ring interconnect permits retrieval notifications to be observed by all potential retrieving interfaces. The ring topology further permits each destination interface to provide feedback to the source interface, |

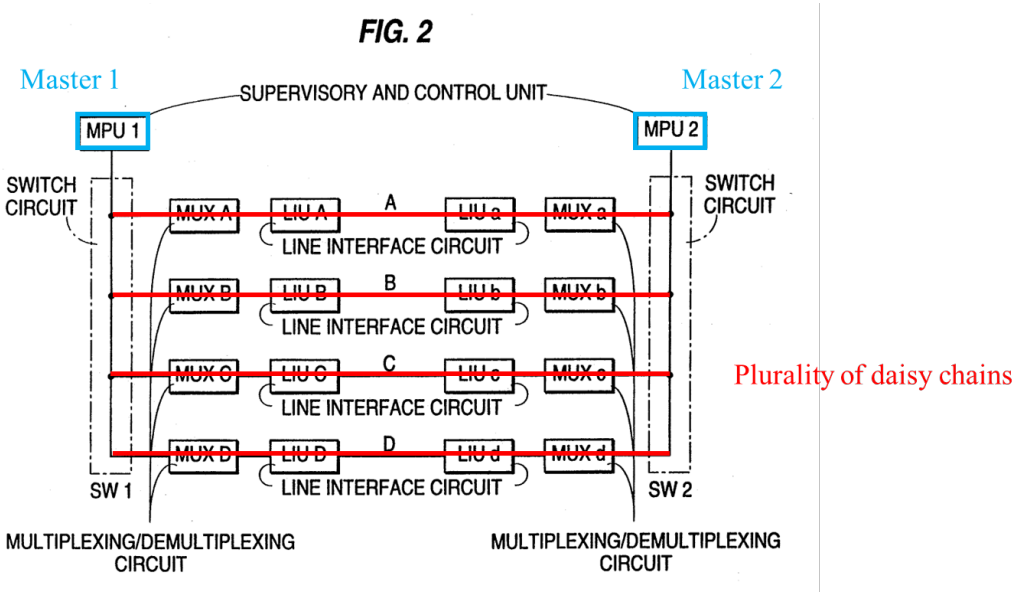
| No. | '904 Patent Claim 14 | The Reference |
|---|---|---|
| | | which is valuable for buffer management applications. Moreover, the point-to-point ring topology bus employs a variable latency access method that enables messages to be passed across the bus with low latency when the system is quiet and with increased latency when the system is busy. In addition, since control messaging around the ring interconnect and across the channel interconnects are embedded in the data stream, the number of pins required and manufacturing costs are reduced." Petersen, 20:38-57. |
| 14[f] | a fabric of one or more switches, which convey the retained packets to the ports, responsive to the address data. | The Reference discloses a fabric of one or more switches, which convey the retained packets to the ports, responsive to the address data.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper. |

| No. | '904 Patent Claim 15 | The Reference |
|---|---|---|
| 15 | The slave unit according to claim 14, wherein the pre-switch is further coupled to receive packets transferred thereto from the second physical interface and to sort the packets in like manner to the packets received through the first physical interface. | The Reference discloses the slave unit according to claim 14, wherein the pre-switch is further coupled to receive packets transferred thereto from the second physical interface and to sort the packets in like manner to the packets received through the first physical interface.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling, and NattKemper. |

| No. | '904 Patent Claim 16 | The Reference |
|---|---|---|
| 16[a] | The slave unit according to claim 15, wherein the retained packets that were received from the first and second physical interfaces and are passed by the pre-switch to the switch fabric are identified by respective first and second port numbers, | The Reference discloses the slave unit according to claim 15, wherein the retained packets that were received from the first and second physical interfaces and are passed by the pre-switch to the switch fabric are identified by respective first and second port numbers.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |
| 16[b] | and wherein in response to a reversal of direction of data flow in the daisy chain, the first and second port numbers are swapped in the pre-switch, so that substantially no reconfiguration of the switch fabric is required in response to the reversal. | The Reference discloses wherein in response to a reversal of direction of data flow in the daisy chain, the first and second port numbers are swapped in the pre-switch, so that substantially no reconfiguration of the switch fabric is required in response to the reversal.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 17 | The Reference |
|---|---|---|
| 17 | The slave unit according to claim 14, wherein when one of the packets received by the pre-switch comprises a multicast packet addressed to one or more of the ports on the slave unit, the pre-switch sorts the multicast packet such that one copy of the packet is retained and another copy of the packet is passed to the second physical interface. | The Reference discloses the slave unit according to claim 14, wherein when one of the packets received by the pre-switch comprises a multicast packet addressed to one or more of the ports on the slave unit, the pre-switch sorts the multicast packet such that one copy of the packet is retained and another copy of the packet is passed to the second physical interface.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 18 | The Reference |
|---|---|---|
| 18 | The slave unit according to claim 14, wherein in the event of a fault in the switch fabric, the pre-switch continues to pass the packets addressed to the succeeding units on to the succeeding units without significant interruption. | The Reference discloses the slave unit according to claim 14, wherein in the event of a fault in the switch fabric, the pre-switch continues to pass the packets addressed to the succeeding units on to the succeeding units without significant interruption.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Below are examples of such references. |

| No. | '904 Patent Claim 18 | The Reference |
|---|---|---|
| | | **Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided.").<br>  <br><br>FIG. 2 (annotation added)<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure |

| No. | '904 Patent Claim 18 | The Reference |
|---|---|---|
| | | of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| 19[preamble] | A method for providing access to a network, comprising: | The Reference discloses a method for providing access to a network, comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |
| 19[a] | coupling first and second master units to interface with the network, | The Reference discloses coupling first and second master units to interface with the network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[a].<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796<br>• Slater '433<br>• Duvvury '626<br>• Duuvury '820 |

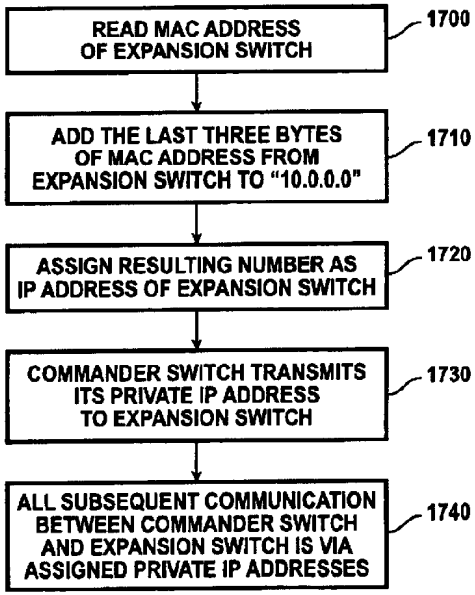
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | **Cisco Catalyst Press Release discloses**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2. |

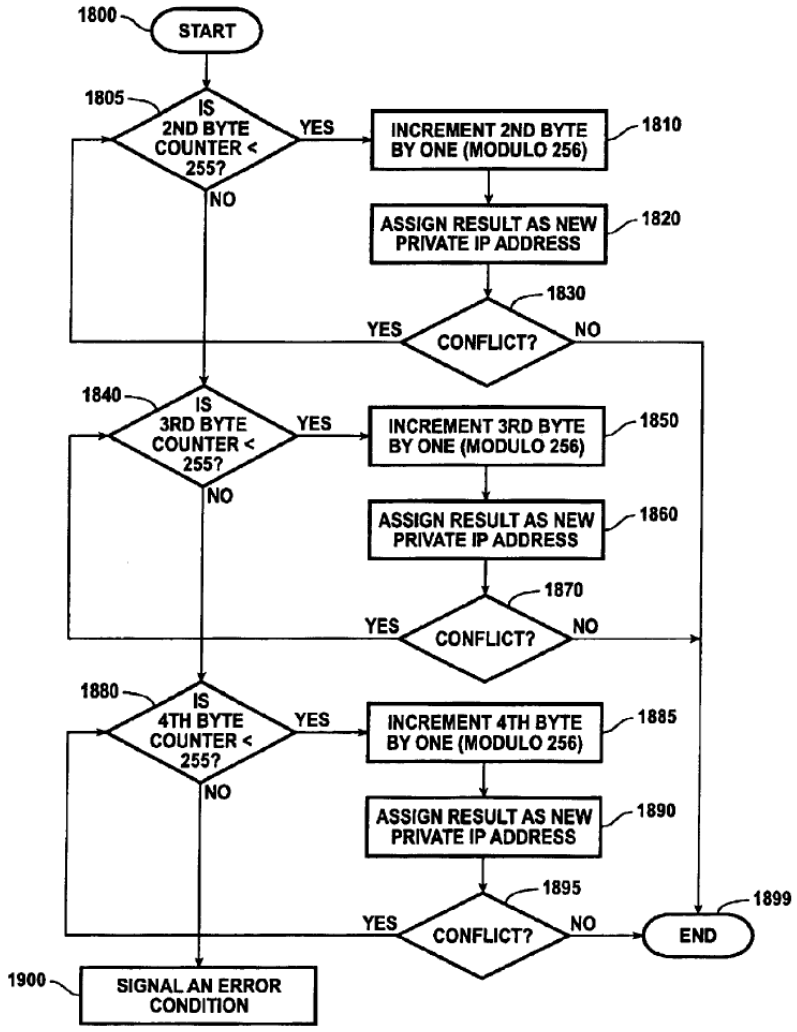
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

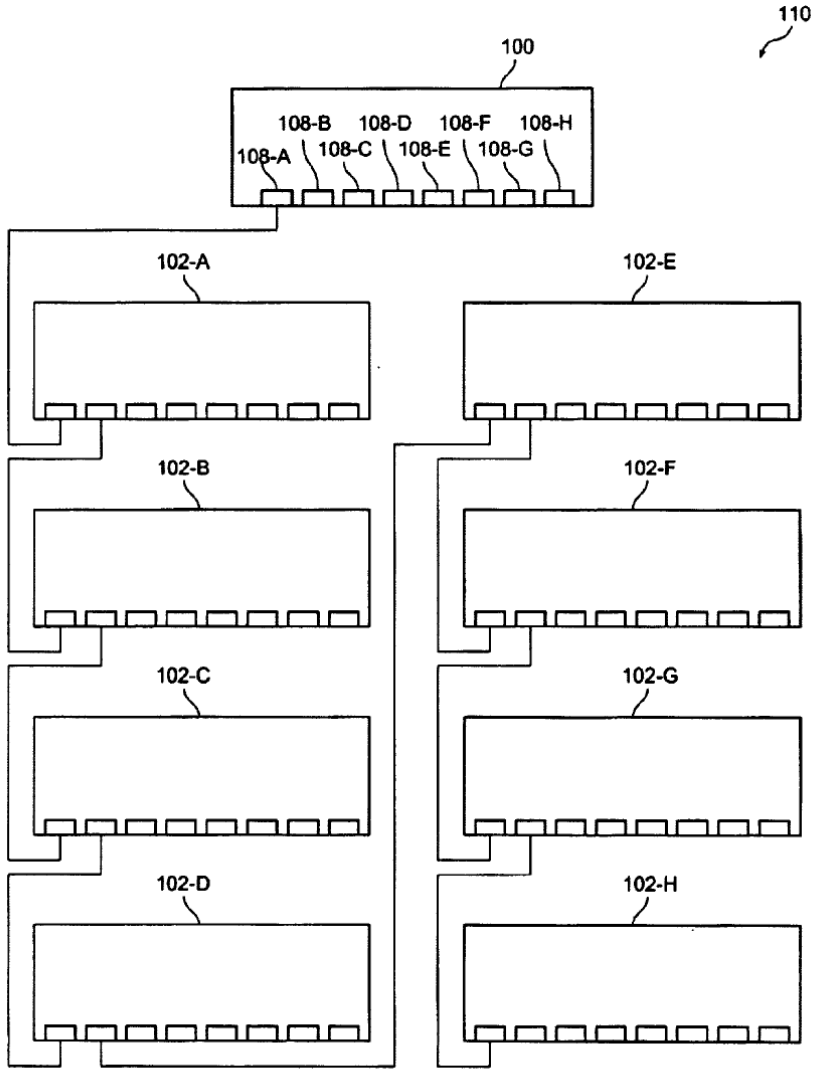
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

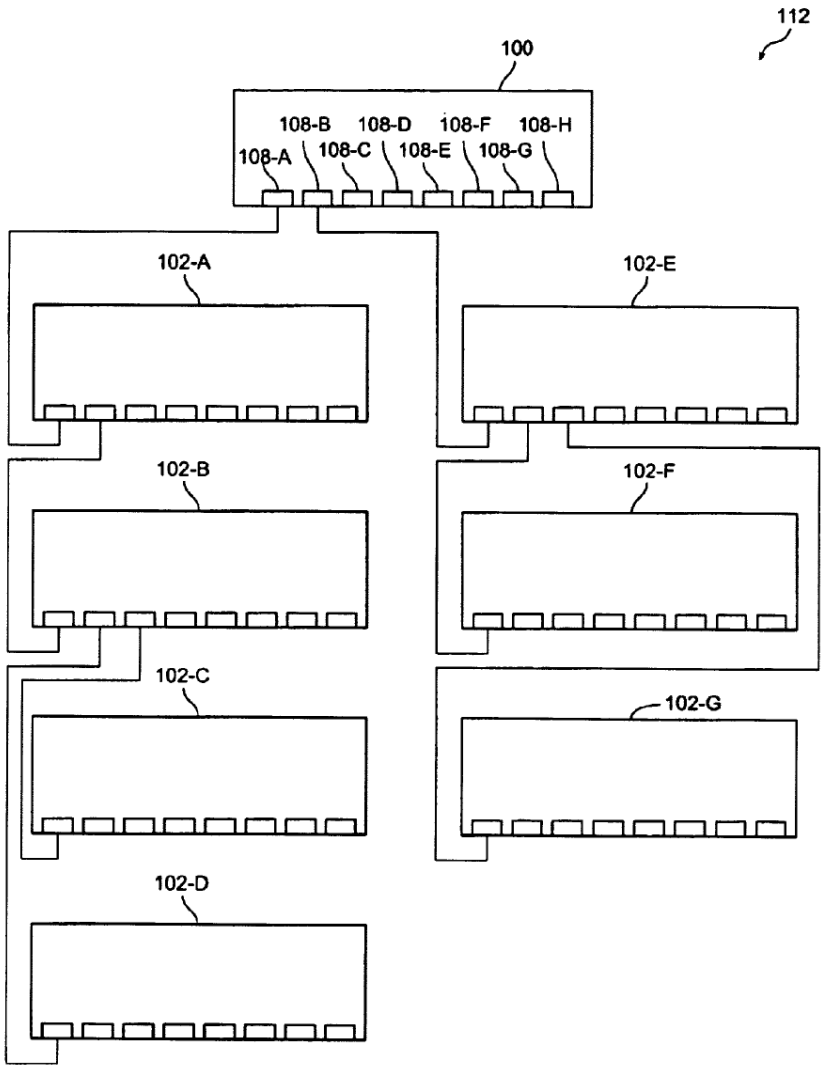
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 19[b] | linking a plurality of slave units, | The Reference discloses linking a plurality of slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, Dowling and Sugawara.<br><br>*See supra* at 1[b], 1[d].<br><br>Cisco already patented "master" and slave" switch technology and commercialized it before Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Cisco Catalyst Press Release<br>• Slater '796 |

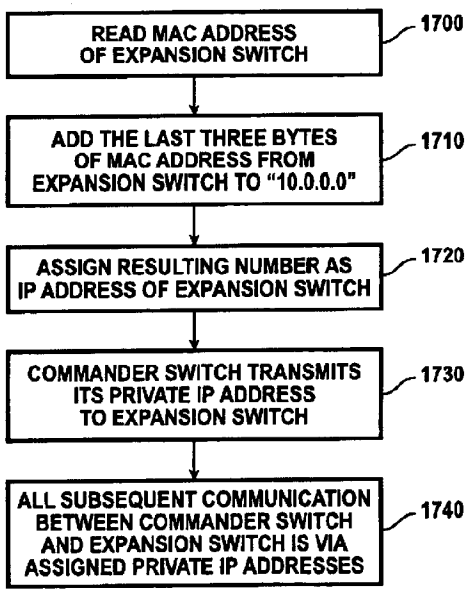
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | • Slater '433<br>• Duvvury '626<br>• Duuvury '820<br><br>**<u>Cisco Catalyst Press Release discloses</u>**:<br>"May 24, 1999 -- Cisco Systems, Inc. today announced the new Catalyst. 3500 Series XL, the industry's most scalable line of stackable 10/100 and Gigabit Ethernet desktop switches that delivers premium performance, manageability, flexibility and unparalleled investment protection." Cisco Catalyst Press Release, 2.<br><br>"The new family of stackable switches, consisting of the Catalyst 3512 XL, Catalyst 3524 XL and Catalyst 3508G XL switches, use Cisco Switch Clustering technology to take traditional stacking to the next level by allowing network managers to manage geographically dispersed switches through a single IP address, using a standard Web browser." Cisco Catalyst Press Release, 2.<br><br>"Cisco delivers next-generation stacking through a new scalable stacking architecture consisting of a new hardware platform, the Gigabit-enabled Catalyst 3500 Series XL; a unique flexible stacking transceiver, the GigaStack Gigabit Interface Connector (GBIC); and Cisco Switch Clustering technology that enables single IP management of geographically dispersed switches." Cisco Catalyst Press Release, 2.<br><br>"The Cisco stacking architecture is fully backwards compatible with all Catalyst 2900 Series XL and Catalyst 1900 Standard and Enterprise Edition switches, giving customers unparalleled flexibility and investment protection." Cisco Catalyst Press Release, 2.<br><br>"'The new Catalyst 3500 XL switches with the unique Cisco Switch Clustering technology and enhanced Cisco Visual Switch Manager makes managing these switches easy and hassle-free,' said Juan Garcia, system network administrator at Acer America. 'You can now manage an entire group of Catalyst 3500 XL, 2900 XL and 1900 switches from a single IP address regardless of their location, using one Web interface. This is a very powerful message to |

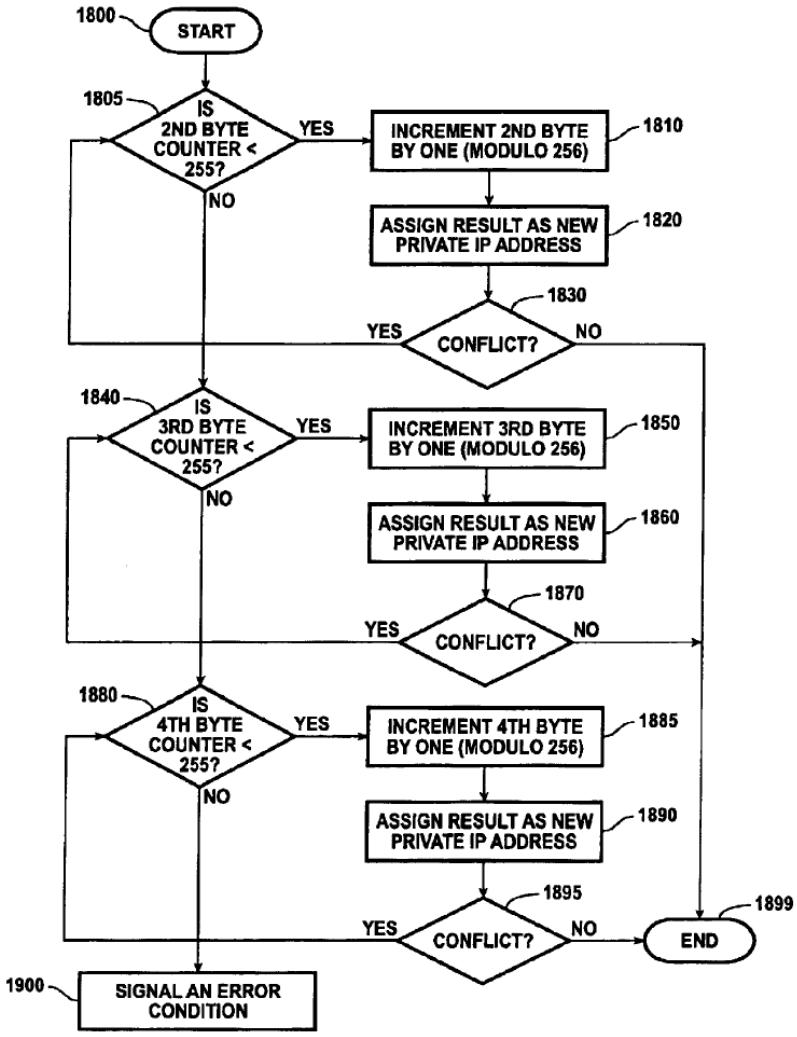
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | network managers that now don't have to manage each individual switch as an independent entity.'" Cisco Catalyst Press Release, 2.<br><br>"With the introduction of the Catalyst 3500 Series XL and Cisco Switch Clustering technology, Cisco introduces next generation stacking. The Catalyst 3500 Series XL switches feature a 10 Gbps switching fabric that delivers wire-speed performance to each 10/100 port. The new stackable switches feature Cisco IOS. software and Cisco Visual Switch Manager (CVSM) software, an easy-to-use, Web-based management interface. All Catalyst 3500 Series XL switches are available in Standard and Enterprise Editions. Enterprise Edition switches offer advanced software features such as, complete 802.1Q and ISL VLAN support, TACACS+ security, and fault tolerance through Uplink Fast." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 Series XL consists of three switch models." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3512 XL: a single rack unit (RU) stackable 10/100 and Gigabit Ethernet switch with 12 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of industry-standard GBICs, including the Cisco GigaStack GBIC, and 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3512 XL is a high-performance, non-blocking switch, ideal for aggregating a small group of Catalyst 2900 XL or Catalyst 1900 switches in a clustered configuration. In a standalone configuration, the Catalyst 3512 XL offers low port density at a low entry price." Cisco Catalyst Press Release, 3.<br><br>"Catalyst 3524 XL: a single RU stackable 10/100 and Gigabit Ethernet switch with 24 10BaseT/100BaseTX ports and two GBIC-based Gigabit Ethernet ports that accommodate a range of GBICs, including the Cisco GigaStack GBIC, 1000BaseSX and 1000BaseLX/LH GBICs. The Catalyst 3524 XL is ideal for delivering dedicated 10 or 100 Mbps bandwidth to individual users and servers in a stack or cluster configuration. Built-in dual GBIC-based Gigabit Ethernet ports provide users with a flexible and scalable solution for Gigabit Ethernet uplinks or GigaStack stacking." Cisco Catalyst Press Release, 3. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Catalyst 3508G XL: a single RU stackable Gigabit Ethernet switch with 8 GBIC-based Gigabit Ethernet ports. The Catalyst 3508G XL is ideal for aggregating a group of 10/100 switches and Gigabit Ethernet servers using Cisco GigaStack GBICs or standard 1000BaseX GBICs." Cisco Catalyst Press Release, 3.<br><br>"The Catalyst 3500 XL and modular Catalyst 2900 XL switches can be stacked using the low-cost Cisco GigaStack GBIC. The two-port GigaStack GBIC offers a range of highly flexible stacking and price/performance connectivity options. It delivers a 1 Gbps stack bus in a daisy-chained configuration or up to 2 Gbps full-duplex connectivity in a dedicated, switch-to-switch configuration. GBIC-based Gigabit Ethernet aggregation via the Catalyst 3508 XL delivers up to 5 Gbps aggregated forwarding bandwidth to connected switches in a switch 'cluster.'" Cisco Catalyst Press Release, 3-4.<br><br>"Cisco Switch Clustering software enables up to 16 interconnected Catalyst 3500 XL, 2900 XL and 1900 switches, regardless of geographic proximity, to form a managed single-IP address network. These switches can be interconnected using a broad range of connectivity options, delivering different levels of performance to meet customer needs. Clustering connectivity options include Ethernet, Fast Ethernet, Fast EtherChannel, low-cost Cisco GigaStack GBIC, Gigabit Ethernet and Gigabit EtherChannel technologies." Cisco Catalyst Press Release, 4.<br><br>"Because the technology is not limited by proprietary stacking modules and stacking cables, Cisco Switch Clustering software expands the traditional stacking domain beyond a single wiring closet and allows users to mix and match interconnections to meet specific management, performance and cost requirements." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering software for the Catalyst 3500 XL, 2900 XL and 1900 switches, enables the management of a 'cluster' of switches through a single IP address. The clusters can be grouped regardless of interconnection media or physical proximity. In a Cisco switch cluster, one Catalyst 3500 XL or 2900 XL switch is designated as the "command" switch and all other switches in the cluster are designated as "member" switches. The command switch |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | serves as the single IP management point and disburses all management action dictated by the network manager." Cisco Catalyst Press Release, 4.<br><br>"Cisco Switch Clustering command software is pre-installed on all Catalyst 3500 XL switches and is available as an upgrade for Catalyst 2900 XL and 1900 switches. Cisco Switch Clustering technology supports Command Line Interface (CLI) in addition to Cisco Visual Switch Manager." Cisco Catalyst Press Release, 4.<br><br>"The Catalyst 3500 Series XL features the Cisco Web-based management tool, Cisco Visual Switch Manager (CVSM 2.0), which allows network managers to view and manage a switch cluster from anywhere on the network through a standard Web browser such as Microsoft Internet Explorer or Netscape Navigator. CVSM 2.0 is launched from the switch itself and delivers simple network and device-level management, including VLAN set-up, port configuration, network cluster views and port monitoring. CVSM is an integral part of the Cisco scalable stacking architecture, allowing users to easily configure and manage switch stacks and clusters, and administer software upgrades across multiple switches." Cisco Catalyst Press Release, 4-5.<br><br>**Duvvury '626 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more member devices. Each network device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have a public IP address. The cluster commander automatically assigns private IP addresses to the other devices in the cluster. Network devices in the cluster constantly monitor network traffic on all their ports to detect conflicts between the automatically assigned IP addresses and the IP addresses of network devices outside of the cluster. When a conflict is detected, the cluster commander assigns a different private IP address to the cluster network device that caused the conflict. The process of detecting and correcting IP address conflicts continues continuously to enable the cluster network devices to react automatically to network configuration changes." Duvvury '626, Abstract. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Duvvury '626, 12:18-30.<br><br>"FIG. 17 is a flow chart illustrating an automatic IP address generation algorithm according to one embodiment of the present invention. When a member switch first joins a cluster, the commander switch generates a CMP address for the member switch by adding last three bytes of the member switch's MAC address to the number "10.0.0.0." Thus, as shown in FIG. 17, at step 1700 the commander switch reads the MAC address of a member switch from an Ethernet frame received from the member switch. Next, at step 1710, the commander switch adds the last three bytes of the member switch's MAC address to the number "10.0.0.0." Then, at step 1720, the commander switch assigns the resulting number to be the CMP IP address of the member switch. For example, if the MAC address of the member switch is "00-e0-1e-01-02-03," then the generated CMIP address will be "10.01.02.03." At step 1730, the commander switch communicates its own CMP address to the member switch. Finally, at step 1740, once a member switch has been assigned a CMP address, the commander switch and the member switch use CMP addresses to communicate with each other." Duvvury '626, 16:11-31. |

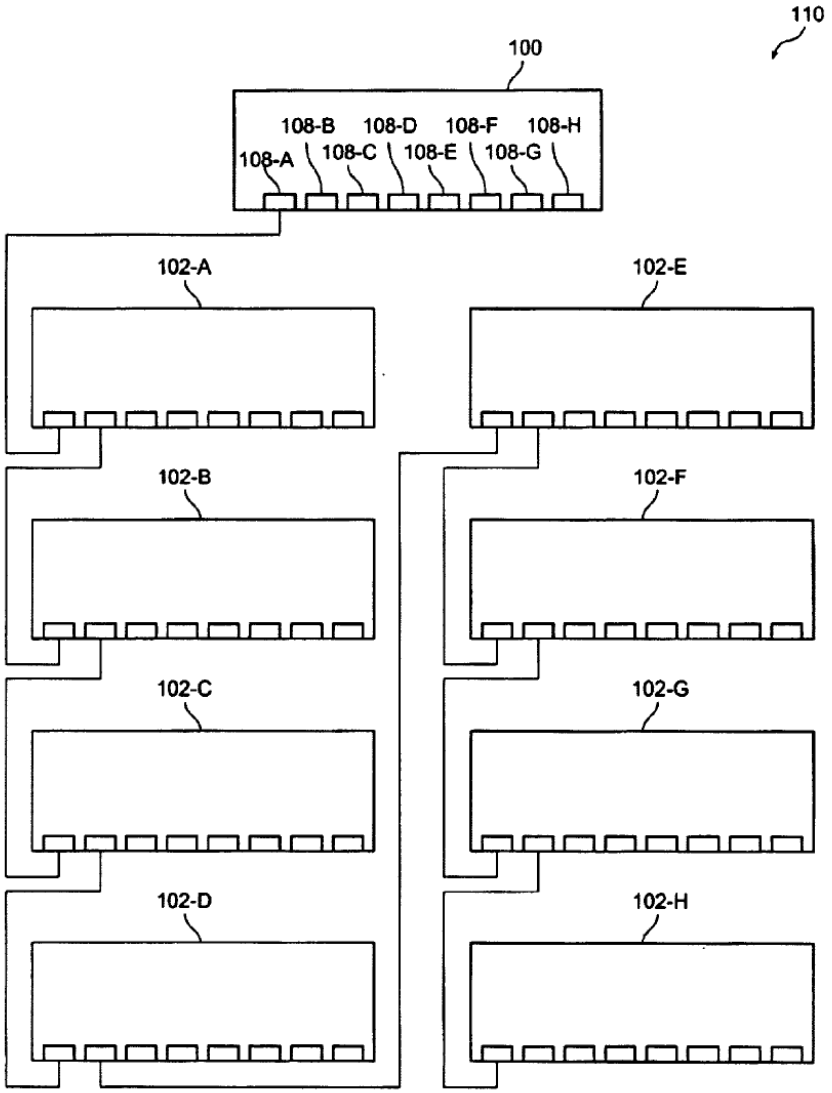
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 17*<br>Duvvury '626, FIG. 17.<br><br>"FIG. 18 is a flow chart illustrating an automatic IP address conflict correction algorithm according to one embodiment of the present invention. In this embodiment, after detecting the conflict, the commander switch generates a new CMP address according to the algorithm shown in FIG. 18. First, at step 1800, three counters are initialized to zero, each representing the number of address correction attempts for the second byte, third byte, and fourth byte of the IP address, respectively. Next, at step 1805, the value of the second byte counter is compared to the highest possible value (255). If the value is less than 255, then at step 1810, the second byte of the IP address is incremented by one, "modulo 256," such that the number wraps back to zero if the present number is 255 and the second byte counter is less than 255. At step 1820, a new CMP address corresponding to the result is assigned to the switch that caused the conflict. At step 1830, if a conflict is still detected, the algorithm loops back to step 1805. Otherwise, the algorithm terminates at step 1899." Duvvury '626, 17:5-16. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 18*<br>Duvvury '626, FIG. 18. |

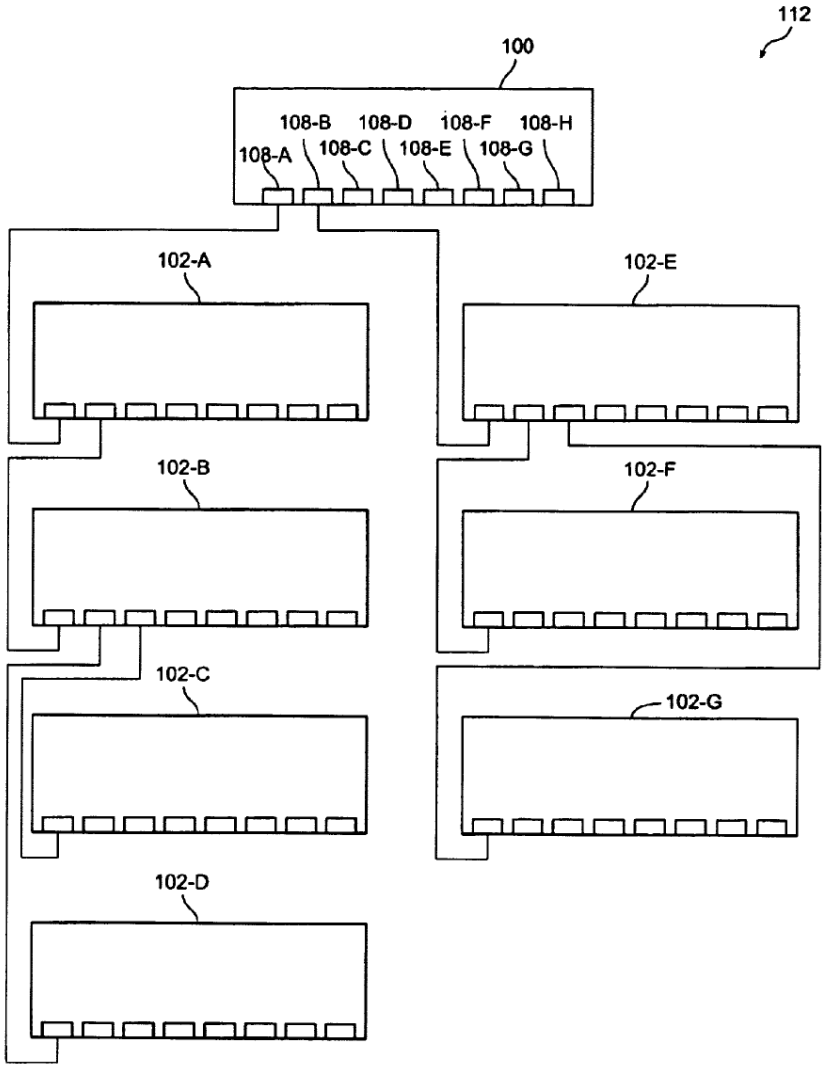
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | **Slater '796 discloses**:<br>"A group of network devices, such as Ethernet switches, are logically configured as a single cluster, with one commander device and one or more expansion devices. Each device in the cluster contains an embedded HTML server that facilitates configuration and management of the network device via a management station running a Web browser. Each device in the cluster is identified by a unique Universal Resource Locator ("URL"). However, only the cluster commander is required to have an IP address. The cluster commander redirects and translates configuration and management requests from the Web browser on the management station so that requests are processed by the appropriate device in the cluster. The exchange of information between the Web browser on the management station and the devices in a cluster is accomplished via redirection of HTTP GET and POST methods. This provides a consistent, device-independent interface between the device and the Web browser on the management station." Slater '796, Abstract.<br><br>"Network devices, such as LAN switches, may be configured and managed using either out-of-band or in-band techniques. Out-of-band configuration and management are typically performed by connecting to the console port on the network device and using the management console locally from a terminal or remotely through a modem. Alternatively, network devices may be configured and managed "in-band," either by connecting via Telnet to the network device and using a management console, or by communicating with the network device's in-band management interface using the industry standard Simple Network Management Protocol ("SNMP"). This can be accomplished by using an SNMP-compatible network management application and the network device's Management Interface Base ("MIB") files. Normally, however, in order to perform in-band administrative tasks of a network device, such as configuration and management, the network device must first be assigned an IP address. Additionally, in order to use in-band configuration and management capabilities, the SNMP management platform of the network device must be configured to understand and be able to access the objects contained in the network device's MIB." Slater '796, 8:55-9:10. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Embodiments of the present invention use a subset of the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite as the underlying mechanism to transport the SNMP configuration and management data. Without limitation, the protocols implemented in embodiments of the present invention include the Internet Protocol ("IP"), the Internet Control Message Protocol ("ICMP"), the User Datagram Protocol ("UDP"), the Trivial File Transfer Protocol ("TFTP"), the Bootstrap Protocol ("BOOTP"), and the Address Resolution Protocol ("ARP")." Slater '796, 9:11-20.<br><br>"The MIB variables of network devices according to embodiments of the present invention are accessible through SNMP. SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. SNMP is used to monitor IP gateways and their networks, and defines a set of variables that the gateway must keep and specifies that all operations on the gateway are a side effect of fetching or storing to data variables. SNMP consists of three parts: a Structure of Management Information ("SMI"), a Management Information Base ("MIB") and the protocol itself. The SMI and MIB define and store the set of managed entities, while SNMP itself conveys information to and from the SMI and the MIB." Slater '796, 9:21-34.<br><br>"Instead of defining a large set of commands, SNMP places all operations in a get-request, get-next-request, and set-request format. For example, an SNMP manager can get a value from an SNMP agent or store a value into that SNMP agent. The SNMP manager can be part of a network management system ("NMS"), and the SNMP agent can reside on a networking device such as a LAN switch. The switch MIB files may be compiled with network management software, which then permits the SNMP agent to respond to MIB-related queries being sent by the NMS." Slater '796, 9:35-44.<br><br>"An example of an NMS is the CiscoWorks™ network management software, available from Cisco Systems, Inc. of San Jose, Calif. CiscoWorks™ uses the switch MIB variables to set device variables and to poll devices on the network for specific information. Among other tasks, the CiscoWorks™ software permits the results of a poll to be displayed as a graph and analyzed in order to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, and monitor traffic loads. Other products known to those |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | of ordinary skill in the art, available from several other vendors, provide similar functionality." Slater '796, 9:45-57.<br><br>"A cluster is a group of connected switches that are managed as a single entity. The switches can be in the same location, or they can be distributed across a network. According to embodiments of the present invention, all communication with cluster switches is through a single IP address assigned to the commander switch. Clusters may be configured in a variety of topologies. As an example, FIG. 8 illustrates a switch cluster 106 configured in a "star," or "radial stack," topology. In this configuration, each of the eight expansion switches 102-A-102-H in cluster 106 is directly connected to one of the ports 108A-108-H of commander switch 100." Slater '796, 10:55-67.<br><br>"A second example of a cluster configuration, known as a "daisy chain" configuration, is shown in FIG. 9. In cluster 110, only expansion switch 102-A is directly connected to the commander switch 100. Expansion switches 102-B-102-G are each connected to an "upstream" switch (one that is fewer "hops" away from commander switch 100) and to a "downstream" switch (one that is more "hops" away from commander switch 100). Finally, the last switch in the chain (expansion switch 102-H) is only connected to its upstream "neighbor" 102-G." Slater '796, 11:1-10. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Slater '796, FIG. 9. |

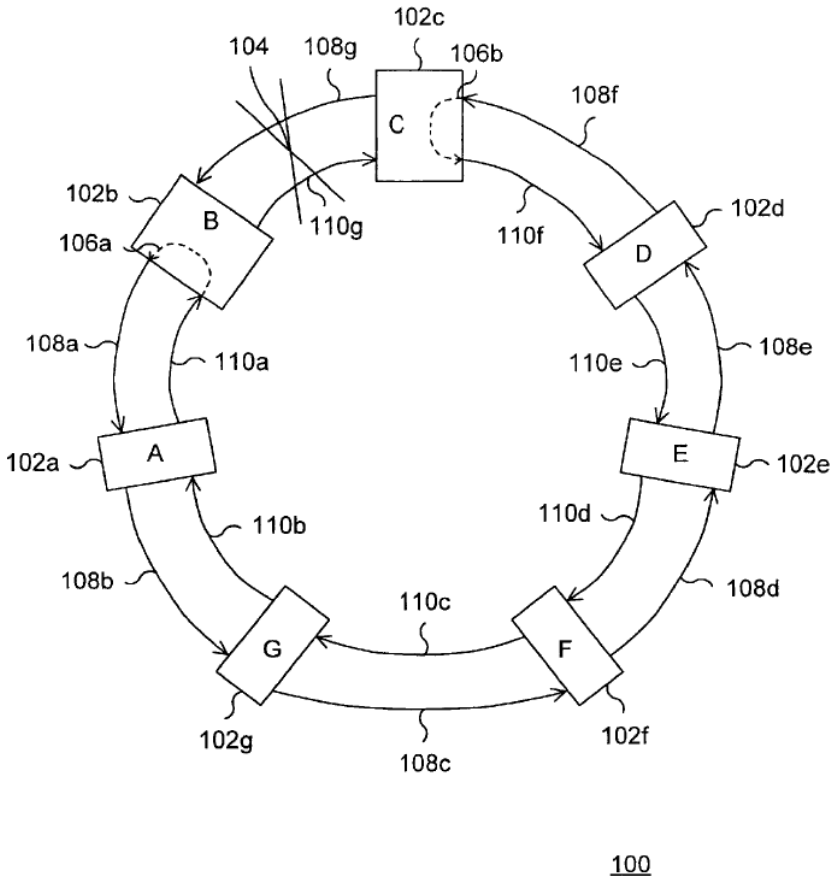
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "As a third example, FIG. 10 illustrates a "hybrid" cluster configuration with one commander switch 100 and seven expansion switches 102-A-102-G. In cluster 112, expansion switches 102-A and 102-E are in a star configuration with respect to commander switch 100. Expansion switch 102-B is in a daisy chain configuration with respect to expansion switch 102-A, while expansion switches 102-C and 102-D are in a star configuration with respect to expansion switch 102-B. Finally, expansion switches 102-F and 102-G are in a star configuration with respect to expansion switch 102-E. Thus, hybrid cluster 112 as shown in FIG. 10 consists of a combination of star and daisy chain configurations." Slater '796, 11:11-22. |

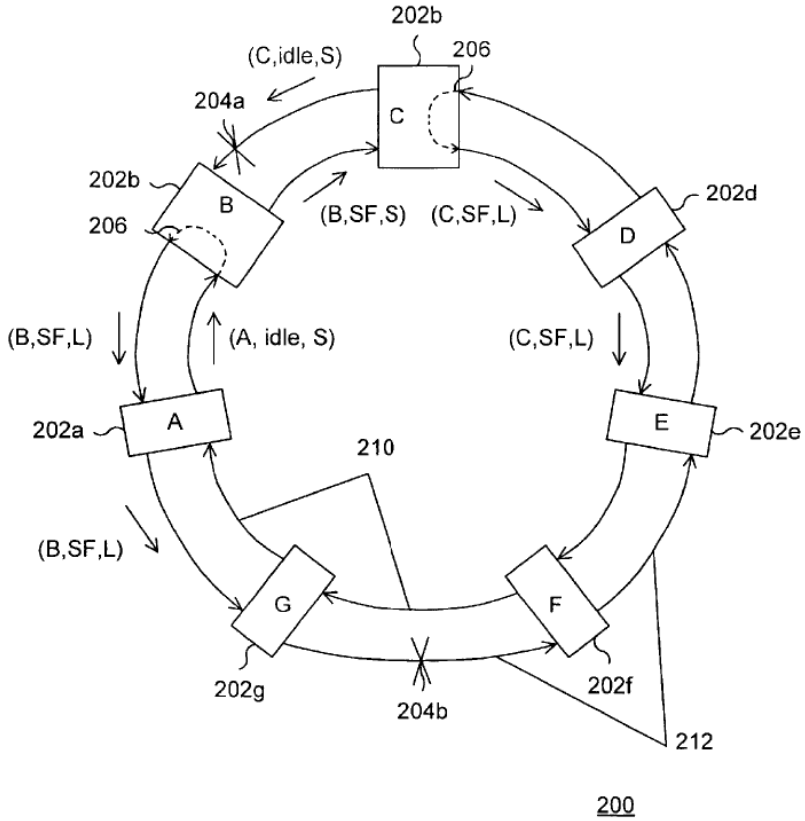
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Slater '796, FIG. 10. |

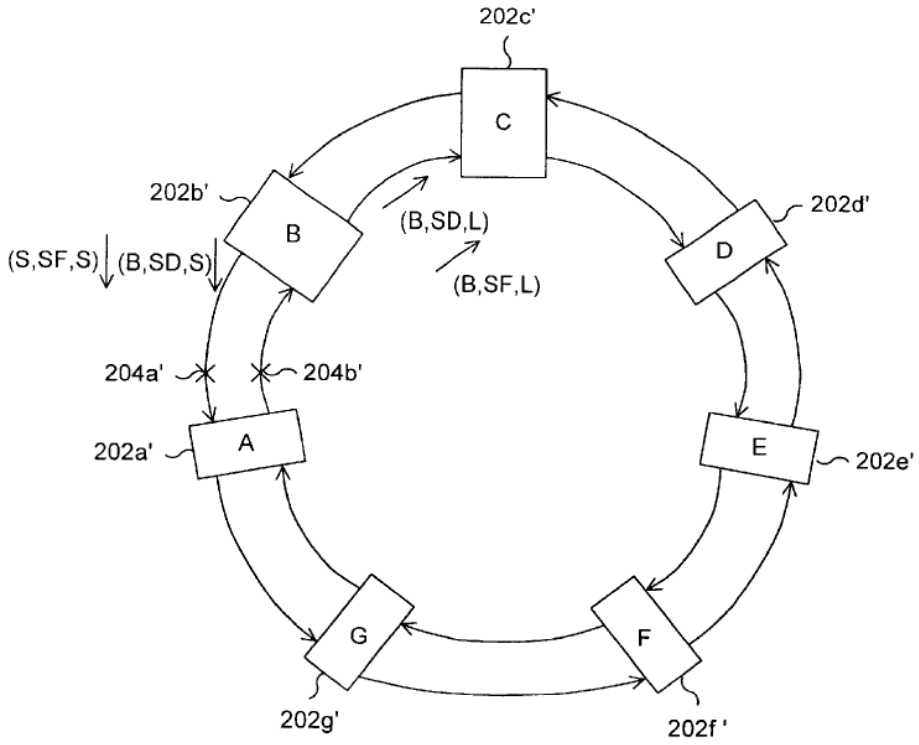
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "The commander switch is the single point of access used to configure and monitor all the switches in a cluster. According to embodiments of the present invention, expansion switches are managed through a commander switch. The commander switch is used to manage the cluster, and is managed directly by the network management station. Expansion switches operate under the control of the commander. While they are a part of a cluster, expansion switches are not managed directly. Rather, requests intended for an expansion switch are first sent to the commander, then forwarded to the appropriate expansion switch in the cluster." Slater '796, 11:26-36.<br><br>"When switches are first installed, they are cabled together according to the network configuration desired for a particular application, and an IP address is assigned to the commander switch. In addition, the commander switch must be enabled as the commander switch of the cluster. Once the commander switch has been enabled, it can use information known about the network topology to identify other network devices in the network that may be added to the cluster. According to one embodiment of the present invention, the commander switch uses the Cisco™ Discovery Protocol ("CDP") to automatically identify candidate network devices. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task. Alternatively, discovery of candidate network devices may be performed manually by inspecting the network topology and the network devices attached to the network." Slater '796, 11:37-54.<br><br>"The method of creating a cluster of Ethernet switches depends on each particular network configuration. If the switches are arranged in a star topology, as in FIG. 8, with the commander switch at the center, all of the expansion switches may be added to the cluster at once. On the other hand, if the switches are connected in a daisy-chain topology, as in FIG. 9, the candidate switch that is connected to the commander switch is added first, and then each subsequent switch in the chain is added as it is discovered by CDP. If switches are daisy-chained off a star topology, as in the exemplary hybrid configuration shown in FIG. 10, all the switches that are directly connected to the commander switch may be added first, and then the daisy-chained switches may be added one at a time." Slater '796, 12:21-34. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "If the commander switch of a cluster fails, member switches continue forwarding but cannot be managed through the commander switch. Member switches retain the ability to be managed through normal standalone means, such as the console-port CLI, and they can be managed through SNMP, HTML, and Telnet after they have been assigned an IP address. Recovery from a failed command switch can be accomplished by replacing the failed unit with a cluster member or another switch. To have a cluster member ready to replace the commander switch, the network administrator must assign an IP address to another cluster member, and know the command-switch enable password for that switch." Slater '796, 12:44-56.<br><br>"One advantage of the present invention is that a network administrator need set only one IP address, one password, and one system SNMP configuration in order to manage an entire cluster of switches. A cluster can be formed from switches located in several different buildings on a campus, and may be linked by fiber optic, Fast Ethernet, or Gigabit Ethernet connections." Slater '796, 13:8-14. |
| 19[c] | each slave unit comprising one or more ports to respective subscriber lines, in a daisy chain between the first and second master units, | The Reference discloses each slave unit comprising one or more ports to respective subscriber lines, in a daisy chain between the first and second master units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* at 1[c]. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| 19[d] | conveying initial downstream data packets, received from the network by one of the master units, along the daisy chain in a first direction, so as to deliver the packets to the ports of the slave units, and | The Reference discloses conveying initial downstream data packets, received from the network by one of the master units, along the daisy chain in a first direction, so as to deliver the packets to the ports of the slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco continues to make innovative contributions to the area of redundant stacked switch technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br><br>Cisco commercialized and patented technology relating to monitoring, detecting, and resolving faults without requiring a network reconfiguration *before* Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Daruwalla<br>• Nederveen<br>• Slater '421<br>• Petersen<br><br>**Daruwalla discloses**:<br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. The present invention provides a protection protocol which simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol will appropriately remove a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, would remove protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, Abstract.<br>  <br><br>Daruwalla, FIG. 1. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 2. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 11.<br><br>"The present invention relates to computer networks. In particular, the present invention relates to a system and method for providing a protection protocol for fault recovery for a two line bi-directional ring network." Daruwalla, 1:8-11.<br><br>"FIG. 1 shows an example of a two line bi-directional ring network. The ring network 100 is shown to include nodes 102 a-102 g. Each node is typically a computer with embedded processors and at least one network connection. Each node 102 a-102 g is shown to be bidirectionally coupled to two neighboring nodes 102 a-102 g via an inner connection ring |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | 110 a-110 g and an outer connection ring 108 a-108 g. For instance, node 102 a is bidirectionally coupled to nodes 102 b and 102 g. The example of FIG. 1 also shows a problem 104 in the connection between node 102 b and node 102 c. When a problem is detected (such as a bi-directional line cut), the connection between nodes 102 b and 102 d wraps back upon itself, as shown by wraps 106 a and 106 b. In this manner, the connection problem 104 can be avoided." Daruwalla, 1:17-30.<br><br>"In a conventional SONET network, each message sent by a sending node to a receiving node typically needs the identification and location of the receiving node to arrive at the proper destination. Accordingly, manual configuration is typically needed in each node to store the identity and location of each other node in the ring network in order to provide for communication between the nodes in the network." Daruwalla, 1:31-44.<br><br>"In summary, for the protection mechanism to operate, each node needs to know the current ring map (current ring topology). What is needed is a system and method for providing fault recovery for two line bi-directional ring network that minimizes the need to keep track of other nodes in the ring network. Preferably, the system would not require reconfiguration of an internal map of the network when a new node is added to, or existing nodes are removed from the network. The present invention addresses such a need." Daruwalla, 2:23-31.<br><br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. According to the embodiment, when the problem is identified, the node sends a message identifying the problem to a second neighbor which is located at least one node away from the problem. The second neighbor then forwards the message to a third neighbor, unless the second neighbor is dealing with a situation that is higher in a hierarchy of situations than the problem described in the message by the original node. In general, if the second neighbor's situation has a higher priority than the situation described by the original node, then the message is ignored and not forwarded. If, however, the message sent by the original node describes a situation with a higher priority than the situation being dealt with by the second neighbor, then, in general, |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | the second neighbor's situation is ignored, at least for the moment, and the original node's message is forwarded to the next neighbor. In general, a higher priority request preempts a lower priority request within the ring. Exceptions are noted as rules of the protection protocol." Daruwalla, 2:35-58.<br><br>"The present invention provides a protection protocol that simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol automatically appropriately removes a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, removes protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, 2:59-3:3.<br><br>"A method according to an embodiment of the present invention for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:4-12.<br><br>"In another aspect of an embodiment of the present invention, a method for adding a new node to a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node prior to an addition of the new node; initiating a fault recovery procedure when the second node receives the first message; receiving a second message from the new node; and entering an idle state when the second message is received." Daruwalla, 3:13-24. |

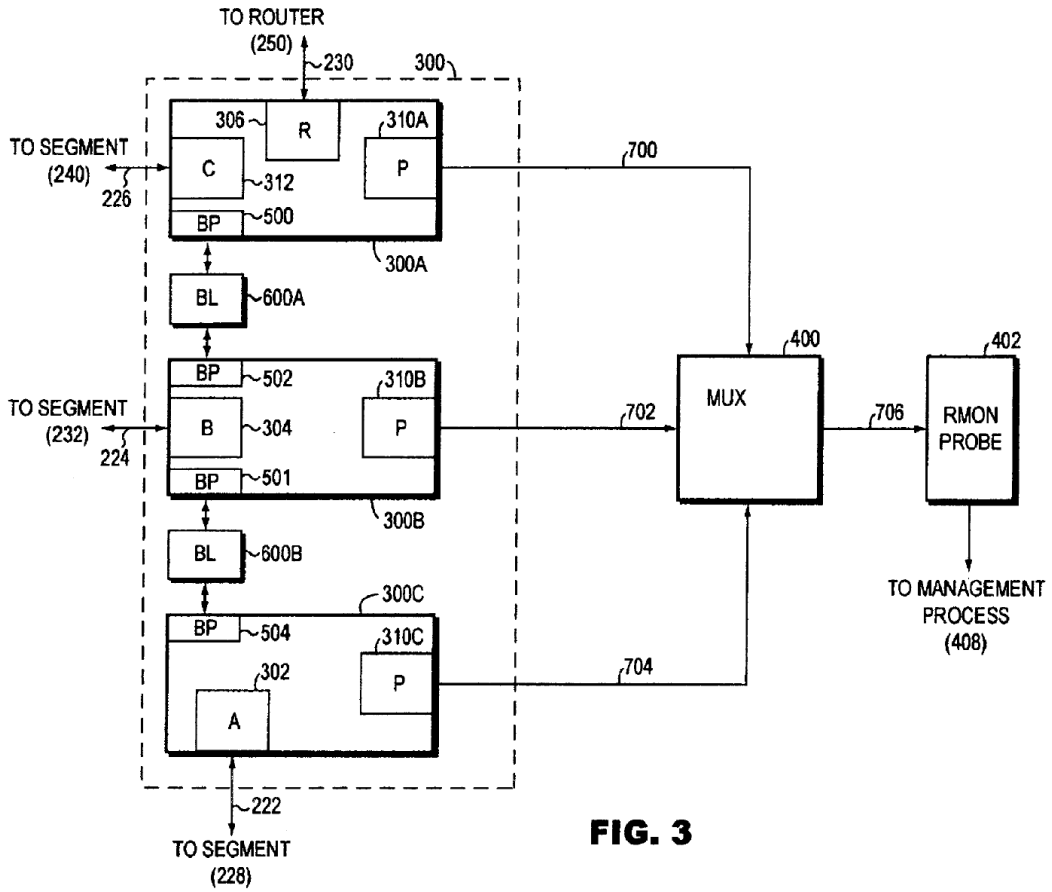
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "In yet another aspect of an embodiment of the present invention, a system for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The system comprises means for detecting a situation by a first node, wherein the first node is one of the plurality of nodes; means for sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and means for initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:25-35.<br><br>"FIG. 2 is block diagram of a ring network utilizing a protection protocol according to an embodiment of the present invention." Daruwalla, 3:40-42.<br><br>"FIGS. 4-6 are flow diagrams illustrating various rules within the protection protocol according to an embodiment of the present invention." Daruwalla, 3:46-48.<br><br>"FIGS. 8-12 are flow diagrams and a system diagram illustrating further rules of the protection protocol according to an embodiment of the present invention." Daruwalla, 3:52-54.<br><br>"FIG. 2 is a block diagram showing a ring network system utilizing a method of fault recovery according to an embodiment of the present invention. The ring network 200 is shown to include nodes 202 a-202 g. The nodes 202 a-202 g are shown to be coupled via an inner ring 210 in which the data flows in one direction, such as a clockwise direction. Additionally, the nodes 202 a-202 g are also shown to be coupled by an outer ring 212 in which data can flow in the opposite direction to the inner ring 210, such as in a counter-clockwise direction. The ring network 200 is shown to have a situation 204 a that requires protection, such as a ring wrap 206." Daruwalla, 5:35-45.<br><br>"FIG. 4 is a flow diagram of an example of a method according to an embodiment of the present invention implied by Rules 1-22. An APS packet is received via step 400. It is determined whether the APS packet has been sent along a long path via step 402. If the packet was not sent via a long path, then the APS packet is not forwarded via step 406. Accordingly, if the APS packet was sent via the short path, then the packet is not forwarded via step 406. If, however, the packet was sent through the long path via step 402, then the APS packet may |

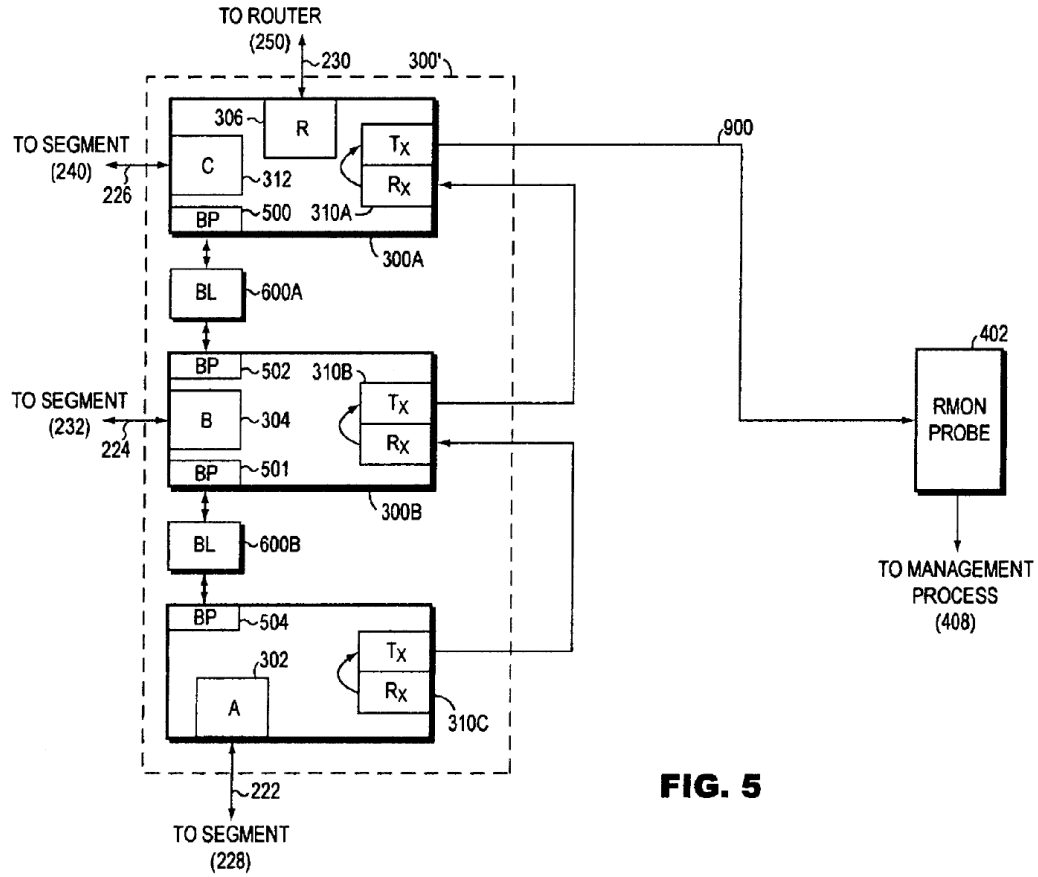
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | be forwarded via step 404. Note that for this example of Rule (1), it is assumed that the long path message does not have to pass through a wrapped connection in order to be forwarded. Otherwise, if the long path message needs to pass through a wrapped connection in order to be forwarded, then the message will simply not be forwarded." Daruwalla, 6:21-36.<br><br>"FIG. 6 is a flow diagram illustrating Rules 4 and 5. A node detects a problem between the node and a first neighbor via step 600. The node performs a wrap away from the side on which the problem exists via step 602. A short path message is then sent to the first neighbor informing it of the problem via step 604. Additionally, a long path message is also sent to a second neighbor informing the second neighbor of the problem via step 604. The first neighbor then performs a wrap away from the side of the problem via step 606. The first neighbor also sends an IDLE message, indicating a wrapped status, on a short path to the node that detected the problem via step 608. This message is sent across the failed span. Note that IDLE messages do not get wrapped and are sent across failed spans if possible. Additionally, the first neighbor also sends a message on a long path toward the side without the problem via step 608." Daruwalla, 6:64-11.<br><br>"An example of the method described in FIG. 6 can be seen in FIG. 2. Node 202 b has detected a problem 204 a and performs a wrap 206 on the side on which the problem exists. Node 202 b also sends a short path message to the neighbor on the other side of the problem 204 a, which is node 202 c. Node 202 b also sends a long path message to its other neighbor node 202 a informing it of the problem. Node 202 c performs a wrap 206 on the side of the problem and sends an IDLE message on a short path to node 202 b. Node 202 c also sent a message on a long path toward the side without the problem to its neighbor 202 d." Daruwalla, 7:12-21.<br><br>"FIG. 7 lists the hierarchy of priorities of Rule (8). For ease of reference, the hierarchy is separated into Class I-III. Class I is the highest priority, while Class III is the lowest priority. An example of a highest priority message in Class I is lockout. Lockout is an order stating that the ring network is not to wrap under any circumstances." Daruwalla, 7:22-26. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Examples of the next priority listed in Class II are forced switch and signal fail. Forced switch indicates that the ring network is configured to wrap at the point of the forced switch. Signal fail is a situation where either two nodes cannot communicate with each other, or one node cannot hear the other node. An example of a signal fail is a physical break in the communication lines between two nodes." Daruwalla, 7:27-33.<br><br>"Note that members of Class II can co-exist (Rule 9). For example, multiple forced switches and signal fails can co-exist. Additionally, members of Class I can co-exist (Rule 10). For example, multiple lockouts in a single ring network can co-exist. However, situations in Class III cannot co-exist with other situations (Rule 11). For example, a signal degrade cannot co-exist with a wait-to-restore." Daruwalla, 7:52-58.<br><br>"When there are multiple requests of the same priority within Class III, the first request to complete a long path signaling will take priority (Rule 13). For example, if there are two signal degrades located on the same ring network, then the first signal degrade which completes the long path signaling will take priority over the other signal degrade. By not allowing members of Class III to co-exist with one another, partitioning of the ring network is avoided." Daruwalla, 7:59-65.<br><br>"In case of two equal requests within Class III on both inner and outer rings of the ring network, the tie is broken by choosing a request on one of the rings, such as the outer ring request (Rule 14). For example, if a signal degrade occurs both on the inner and outer rings, then a request on a predetermined ring, such as the outer ring, will take priority over the other requests." Daruwalla, 7:66-8:5.<br><br>"FIG. 8 is a flow diagram illustrating Rules (9), (10), (11), (13), and (15). Note that the flow diagram described in FIG. 8 is merely an example of one way in which the rules of the method according to the embodiment of the present invention can be executed. For example, the determination of whether the long path message is a Class I request via step 802 or a Class II request via step 810 can be in reverse order." Daruwalla, 8:6-11. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "A wrapped node receives a long path message via step 800. It is then determined if the long path message is a Class I request via step 802. The classes used in FIG. 8 are meant to correspond with the example of classes defined in FIG. 7. If the long path message is a Class I request, then it is determined if a local situation also has a Class I request via step 804. A local situation includes scenarios such as when a node detects a situation or problem, or when a node is made aware of a problem or situation via a short path message from its neighbor. If a local situation is not a Class I request via step 804, then any existing local wraps are unwrapped and the long path message is forwarded via step 806. If, however, a local situation is a Class I request via step 804, then the connections are already unwrapped or was never wrapped, and the long path message is forwarded via step 808." Daruwalla, 8:12-26.<br><br>"FIG. 12 is a flow diagram illustrating rules (20) and (21) of the method according to the embodiment of the present invention. A wrapped node determines that a problem has been cleared via step 1200. It then enters a wait-to-restore state via step 1202. It is then determined if its neighbor is the same neighbor as previously noted via step 1204. The node can save the source of a short path message at the time of wrap initiation to note the identity of its neighbor. If the current neighbor is not the same as the previous neighbor via step 1204, then an IDLE state is entered via step 1206. If, however, the current neighbor is the same as the previous neighbor via step 1204, then it is determined whether a pre-determined wait-to-restore time has expired via step 1208. Once the pre-determined wait-to-restore time has expired, then the node enters an IDLE state via step 1206." Daruwalla, 12:60-13:6.<br><br>"A method and system for fault recovery for a two line bi-directional network has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor." Daruwalla, 13:7-19. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | **Nederveen discloses**:<br>"A technique for use in gathering network activity-related information from cascaded network switches is provided. Using this technique, the information can be gathered without substantially reducing performance of the cascaded switches. In one embodiment, a single remote monitoring probe is connected via respective connections to each of the switches so as to receive the information from the switches. In another embodiment, only one of the switches is connected to the probe, and the other switches transmit their respective portions of the information to the switch connected to probe. The switch connected to the probe provides these portions of the information, as well as, any of its respective activity-related information to the probe. In this latter embodiment, the switches may be connected by dedicated connections and switch ports that are used solely for communicating the activity-related information." Nederveen, Abstract. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>**FIG. 3**<br>Nederveen, FIG. 3. |

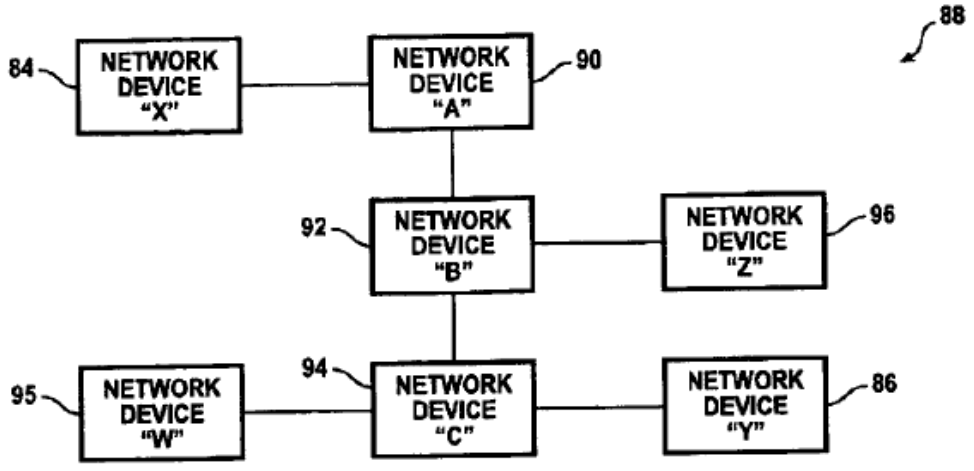
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>**FIG. 5**<br><br>Nederveen, FIG. 5.<br><br>"Thus, it would be desirable to provide a stacked switch monitoring technique that permits efficient offloading of raw data processing from the stacked switches, requires only a minimal number of specialized network entities to gather and process such raw data, and does not result in substantial degradation of stacked switch performance." Nederveen, 4:38-43. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Accordingly, the present invention provides a technique for remote monitoring of a switch network that overcomes the aforesaid and other disadvantages and drawbacks of the prior art. More specifically, in one aspect of the present invention, a technique is provided for gathering information that may be useful in network management (e.g., switch port activity-related information), from switches in the network that are in a stacked configuration. The information is gathered from the stacked switches by a single network entity (e.g., an SNMP remote monitoring probe) in such a way that it does not substantially degrade the performance of the switches. This is accomplished, in one embodiment of the technique of the present invention, by connecting the switches via respective connections to a multiplexer that selectively connects the switches, according to an arbitration scheme, to the single network entity. The entity gathers respective portions of the information from switches when it is connected to the switches by the multiplexer. The information gathered by the entity may be provided to another network entity (e.g., an SNMP management node) in order to permit the other entity to use that information in managing the network." Nederveen, 4:46-67.<br><br>"In another embodiment of the technique of the present invention, only one of the switches is connected to the single information gathering entity. The switches that are not connected to the entity transmit, via respective dedicated ports and connections (i.e., ports and connections that are used solely for network information gathering activities), their respective portions of the information to the switch that is connected to the entity. The switch that is connected to the entity transmits, via a respective dedicated port and connection, the information received from the other switches, as well as, its own information to the entity." Nederveen, 5:1-11.<br><br>"FIG. 3 is a schematic, functional block diagram illustrating in greater detail the construction of the stacked switch network shown in FIG. 2." Nederveen, 5:26-28.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network configured to employ another embodiment of the present invention." Nederveen, 5:32-34. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "FIGS. 2-5 illustrate features of a computer network 200 wherein embodiments of the present invention may be advantageously practiced. Network 200 comprises a stacked switch network 300 which interconnects a plurality of network segments 228, 232, 240, and 251. Each segment 228, 232, 240 comprises one or more local area networks having computer endstations (not shown). Segment 251 is a network router segment that comprises network router 250. Each segment 228, 232, 240 is coupled via a respective communications link 222, 224, 226 to a respective port 302 (i.e., port A), 304 (i.e., port B), 312 (i.e., port C) of the switch network 300. Likewise, the router 250 of router segment 251 is coupled via a respective trunk line 230 to router port 306 (i.e., port R)." Nederveen, 5:46-59.<br><br>"Stacked switch network 300 comprises a plurality of data network switches 300A, 300B, 300C (e.g., Catalyst 3900™ series switches of the type commercially available from the Assignee of the subject application) coupled together via conventional stack link bus connection logic 600A, 600B. More specifically, logic 600A couples a stack link bus port and associated logic 500 in switch 300A to a stack link bus link port and associated logic 502 in switch 300B. Similarly, logic 600B couples another stack link bus port and associated logic 501 in switch 300B to a stack link bus port and associated logic 504 in switch 300C. It should be understood that although, as is shown in FIG. 3, switches 300A and 300B, and switches 300B and 300C, may be coupled serially together by separate respective logic elements 600A, 600B, each of the switches 300A, 300B, 300C may be coupled together via a single respective stack link bus port in the switch to a single stack link bus connection logic block (not shown, e.g., of the type that is commercially available under the tradename Catalyst Matrix™ from the Assignee of the subject application). Further alternatively, depending upon the particular design and functionality of the ports 500, 501, 502, and 504, and the control and forwarding logic (whose operation will be described more fully below) in the switches 300A, 300B, 300C, the circuitry in logic 600A, 600B may instead be comprised in the ports 500, 501, 502, and 504 and/or control and forwarding logic, and therefore, in this alternative configuration, the logic 600A, 600B in the network 300 may be replaced by simple connection means (e.g., cable connectors)." Nederveen, 6:29-57. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Each switch 300A, 300B, 300C includes a respective internal bus (e.g., element 800 in switch 300C) that is coupled via at least one stack link bus port and associated interface logic (e.g., 504 in switch 300C) to external stack link bus connection logic (e.g., element 600B in switch 300C). Each switch 300A, 300B, 300C also includes respective programmable control and forwarding logic (e.g., element 802 in switch 300C) comprising processing, memory, and other circuitry for storing and learning configuration information (e.g., source and destination MAC addresses of messages received by the switch, switch bridging table, switch segments' spanning tree and virtual local area network information, etc.), and for providing appropriate commands to other elements (e.g., the switch ports) to cause data messages received by the switch to be forwarded to appropriate network segments coupled to the switch based upon this configuration information. In each switch, the switch's port logic circuitry (e.g., port A logic 302 and port P logic 310C in switch 300C) and control and forwarding logic are coupled to each other via that switch's respective internal bus. The stack link bus port and associated logic in each switch 300A, 300B, 300C may comprise a Catalyst™ stack port line interface card (commercially available from the Assignee of the subject application) inserted into a bus expansion slot (not shown) in the switch. Although not shown in the Figures for purposes of clarity of illustration, each switch 300A, 300B, 300C in network 300 typically will include tens or hundreds of ports coupled to network segments." Nederveen, 6:58-7:19.<br><br>"The control and forwarding logic and stack link bus port and associated logic in each switch, and the logic 600A, 600B, are configured to together implement conventional techniques for permitting the switches 300A, 300B, 300C to function together as a single logical/virtual switch. More specifically, when configured in the stacked arrangement 300, after the switches 300A, 300B, 300C and logic 600A, 600B are initially activated, they execute initial power-on self-diagnostics, and thereafter, enter a "stack discovery" mode of operation." Nederveen, 7:20-29.<br><br>"In the stack discovery mode of operation, the control and forwarding logic in each switch 300A, 300B, 300C first "senses" that its switch is coupled to logic 600A and/or 600B, and then determines the particular configuration of the stacked switch network 300, using suitable conventional autosensing/autoconfiguration techniques. The control and 321orwarding logic in the switches 300A, 300B, 300C then assigns to the switches respective unique |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | identification numbers (e.g., based upon unique identification numbers of respective ports of the logic 600A, 600B to which the switches are coupled)." Nederveen, 7:30-40.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network 300′ configured to employ another embodiment of the present invention. It should be understood that unless specifically stated to the contrary, the structure and operation of the network 300′ are substantially the same as the structure and operation of network 300. In network 300′, each of the dedicated ports 310A, 310B, 310C comprises a respective transmit portion and receive portion, referenced in FIG. 5 as RX and TX, respectively." Nederveen, 11:20-29.<br><br>**Slater '421 discloses**:<br>"A method and apparatus for discovering paths to other network devices includes a protocol and network management application that can be executed on network devices. The Ethernet protocol is used to detects paths to other network devices, knowing only the Ethernet address of the destination. A discovery protocol is extended to add hop probe and hop probe reply Type-Length-Value fields in a variable-length list. The hop probe fields contain a hop count, a destination Ethernet address, and a source Ethernet address. When a hop probe is received by a network device, the hop count field is decremented by one and the hop probe is forwarded. Packet received with a hop count of one are not forwarded and a hop probe reply is sent back to the Ethernet source address of the hop probe. The hop probe reply fields contain a destination Etherned address and a source Ethernet address." Slater '421, Abstract. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br><br>Slater '421, FIG. 6.<br><br>"Partly as a result of the increased complexity of networks, network administrators must often troubleshoot problems with their network. Two classes of network problems often faced by network administrators are "reachability" problems and performance slowdowns. Reachability problems occur when one or more network devices cannot be accessed through a network, and can be caused by hardware or software failures, cabling problems, or any of several other types of difficult-to-diagnose problems that can occur in a network." Slater '421, 7:11-20.<br><br>"Some of the world's largest networks today rely on the TCP/IP suite of networking protocols. With a relatively small kit of basic tools, network administrators can learn much about an internetwork. "Ping" and "traceroute" commands, "show" commands, and "debug" commands (all of which are typically available via the basic management interface on a network device) form the core of the network administrator's internetwork toolkit. Ping and |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | traceroute commands can be useful tools in determining where failures are occurring, but they are cumbersome to use, and require knowledge of the IP address or host name of the destination network device. The show commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data. Finally, debug commands provide clues about the status of a network device and other network devices directly or indirectly connected to it. Because debug commands can create performance slowdowns, they must be used with great care, and using the wrong debug command at the wrong time can exacerbate problems in already poorly performing networks." Slater '421, 7:55-8:8.<br><br>"Embodiments of the present invention as illustrated herein use the Cisco™ Discovery Protocol ("CDP") to automatically detect paths to specified network devices in Ethernet LANs. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task." Slater '421, 9:10-15.<br><br>"CDP is a media-independent device discovery protocol which can be used by a network administrator to view information about other network devices directly attached to a particular network device. In addition, network management applications can retrieve the device type and SNMP-agent address of neighboring network devices. This enables applications to send SNMP queries to neighboring devices. CDP thus allows network management applications to discover devices that are neighbors of already known devices, such as neighbors running lower-layer, transparent protocols." Slater '421, 9:16-26.<br><br>"It is to be understood that the present invention is not limited to devices that are compatible with CDP. CDP runs on all media that support the Subnetwork Access Protocol ("SNAP"), including LAN and Frame Relay. CDP runs over the data link layer only. Each network device sends periodic messages to a multicast address and listens to the periodic messages sent by others in order to learn about neighboring devices and determine when their interfaces to the media go up or down. Each device also advertises at least one address at which it can receive SNMP messages. CDP messages, or "advertisements," contain holdtime information, which indicates the period of time a receiving device should hold CDP information from a neighbor before discarding it. With CDP, network management applications can learn the device type |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | and the SNMP-agent address of neighboring devices. This process enables applications to send SNMP queries to neighboring devices." Slater '421, 9:27-43.<br><br>"It should be noted that-normally, CDP packets according to aspects of the present invention are transmitted at regular intervals (e.g. once every 60 seconds). However, in embodiments of the present invention, when a Hop Probe or Hop Probe Reply needs to be forwarded by a network device, the network device is commanded to send a CDP packet immediately." Slater '421, 16:66-17:5.<br><br>"The present invention is much faster than the previous method that involved logging in to each intermediate network device, entering the "show cdp neighbors" command, and interpreting the output to find the next hop along the path to the destination network device. Also, the present invention allows individual users, such as network administrators, to execute a tool to manually discover paths through a network of Ethernet switches. The present invention can be used by network management software to automatically map the topology of clusters of network devices, such as Ethernet switches. Finally, the present invention is useful in loop detection. Enhancements to Spanning Tree and other bridge-level routing protocols can test proposed changes to switch topology prior to making them." Slater '421, 17:6-20.<br><br>**Petersen discloses**:<br>"Methods and apparatus for enabling communication between a source network device and one or more destination network devices are disclosed. A system enabling communication between a source network device and one or more destination network devices includes a switch and a ring interconnect. The switch is adapted for connecting to the source network device and the one or more destination network devices. More particularly, the switch is capable of storing data provided by the source network device and retrieving the data for the one or more destination network devices. The ring interconnect is adapted for connecting the source network device and the one or more destination network devices to one another. In addition, the ring interconnect is capable of passing one or more free slot symbols along the ring interconnect. Thus, the ring interconnect is capable of expanding when one or more of the free slot symbols are each replaced by a frame notify message indicating that the data has |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | been stored by the switch for retrieval by the one or more destination network devices." Petersen, Abstract. |

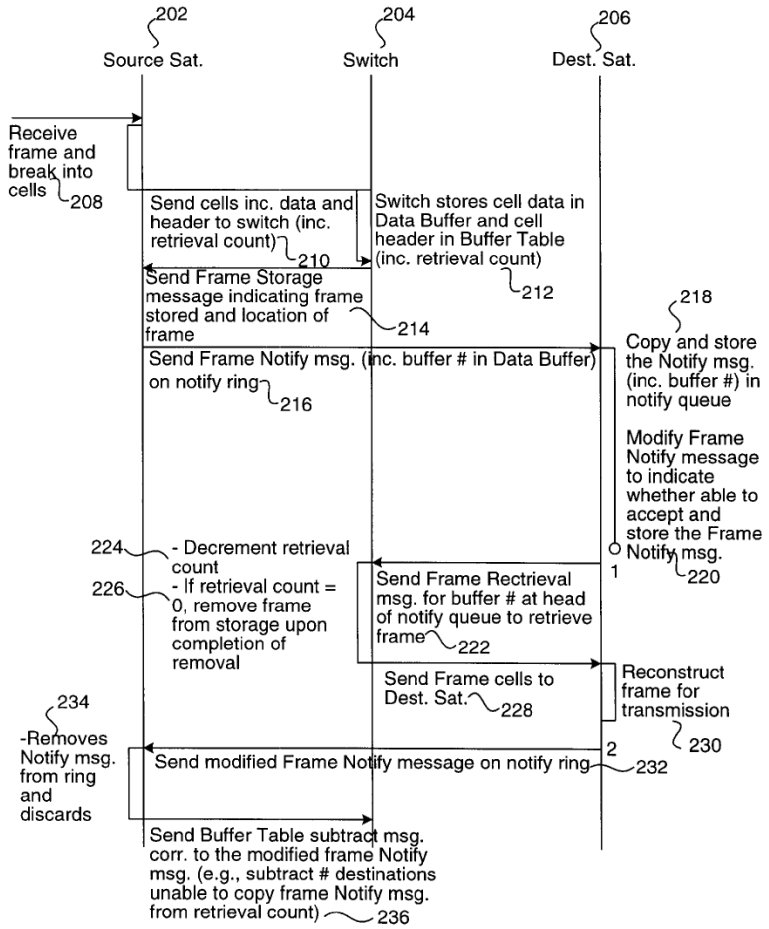



***FIG. 2***

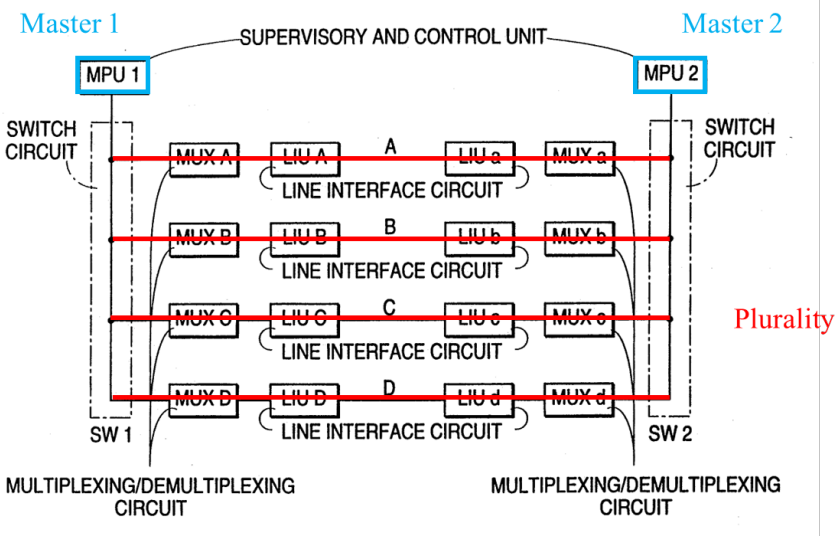
Petersen, FIG. 2.

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "The present invention relates to a mixed topology data switching system that combines a radial interconnect with a ring interconnect. More particularly, the radial interconnect permits devices to store and retrieve data using a switch, while the ring interconnect permits devices along the ring interconnect to provide notification that data has been stored for retrieval, as well as provide feedback regarding the ability or inability to retrieve such data." Petersen, 1:34-41.<br><br>"In controlling the flow of network traffic through a switching system, it is often desirable to provide feedback to the source of the data. For instance, although a transmitting device, hereinafter referred to as a "source device," may transmit or forward data to a receiving device, hereinafter referred to as a "destination device," the destination device may be incapable of handling the data. In these circumstances, the source device is often unaware that the data was not accepted by the destination device, complicating switch management. Common solutions to the problem of switch traffic management have included ensuring that all intended destination devices are "ready to receive" prior to transmitting data on a ring or bus interconnect, or insisting that each intended destination device send an explicit acknowledgement back to the source device. Both of these approaches result in reduced efficiency of the interconnect scheme. By way of example, in a ring network, such acknowledgment is typically provided in the data frame being transmitted. As another example, in other interconnect schemes, each such device may send a separate acknowledgment, therefore adding to the traffic on the network. Accordingly, it would be desirable if a traffic management scheme were established which could provide such feedback to the source of the data while minimizing traffic management overhead." Petersen, 2:8-32.<br><br>"According to one embodiment, the present invention combines the use of two data transport methods: a point-to-point radial interconnect and a ring interconnect. The radial interconnect connects interface devices to each other through the services of a central switch device to permit the transport of data. Typically, a single interface has a single dedicated radial interconnect to the central switch. These interface devices are further connected to one another via a ring interconnect to convey retrieval notifications regarding forwarding of the data (by source devices) and receipt of the data (by destination devices)." Petersen, 2:36-46. |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | "Each radial interconnect provides a narrow, high baud-rate connection to convey to the actual data from and to the associated interface without being burdened by the unrelated traffic for the remaining interfaces in the system. This is accomplished through the use of a central switch device, which stores and retrieves data for the various interfaces in the system. As will be apparent from the following description, this architecture provides numerous advantages over a wide parallel bus or ring." Petersen, 2:47-55.<br><br>"The ring interconnect may be used to convey a "retrieval notification"(i.e., retrieval message) that may be observed by all potential retrieving interfaces. The retrieval notification notifies specific devices ("destination devices") or interfaces that one or more frames addressed to them are available from the switch device. Moreover, the ring interconnect permits each destination device to provide feedback to the source device letting the source know whether the destination has accepted the notification provided by the source device and therefore whether the destination can retrieve the data intended for it. The feedback is particularly useful in buffer management applications. In this manner, an efficient and flexible data transport and retrieval notification system that includes a feedback path to the source of the data is provided." Petersen, 2:56-3:3.<br><br>"FIG. 2 is a process flow diagram illustrating a method of providing network communication according to an embodiment of the invention." Petersen, 3:9-11.<br><br>"FIG. 2 is a process flow diagram illustrating in further detail a method of providing network communication in the above-described system according to an embodiment of the invention. As shown, process steps performed by a source device 202 are illustrated along an associated vertical line, steps performed by a switch 204 are illustrated along another vertical line, and steps performed by a destination device 206 are illustrated along still another vertical line." Petersen, 4:50-57.<br><br>"When the frame is stored by the switch 418, the source device preferably receives an acknowledgment that the data has been stored. Thus, to provide this feedback, a frame storage message (i.e., storage reply) is sent from the switch 418 on the channel 416 to the channel interface 414. The frame storage message is then provided to the notify ring interface as |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | shown at 430 and sent on the notify ring. Once this acknowledgment is received by the interface device 402, the designated destination devices may be notified via notify ring interface 424. As described above, a Frame Notify message may be sent via the notify ring interface 424 to the destination devices. More particularly, the Frame Notify message may identify one or more destination devices for the frame and specify the location of the frame to be retrieved. By way of example, the location of the frame to be retrieved by the destination devices may be designated by a buffer number 430. In addition, the destination devices for the frame may be specified in the Frame Notify message through a notify queue map 426. More particularly, the notify queue map 426 may specify a notify queue associated with a particular destination device. The notify queue may be expressly designated through the use of one or more bits as well as implied through the specification of a priority level for the data. The notify queue map 426 will be described in further detail with reference to FIG. 13. The notify ring interface 424 then creates a Frame Notify message including the notify queue map 426 and the buffer number 430 which is then sent on an outbound interface of the notify ring 432." Petersen, 7:55-8:16.<br><br>"As described above, the notify ring may be expanded to accommodate communication between interface devices. The communication between the interface devices and the switch is therefore performed on one or more channels rather than the notify ring. As a result, the flexibility of the notify ring does not effect the speed with which the interface devices may communicate with the switch. Thus, where a single port operates at a faster speed than the channels, multiple channels may be grouped together. In this manner, the speed with which the switch may communicate with the interface devices may be maximized." Petersen, 20:27-36.<br><br>"The present invention provides a mixed topology data switching system that combines a point-to-point radial interconnect with a ring interconnect to maximize the speed of network traffic. The radial interconnect provides a narrow, high baud-rate connection to convey the data traffic for just the interface in question, without being burdened by all of the unrelated traffic for the remaining interfaces in the system. At the same time, the ring interconnect permits retrieval notifications to be observed by all potential retrieving interfaces. The ring topology further permits each destination interface to provide feedback to the source interface, |

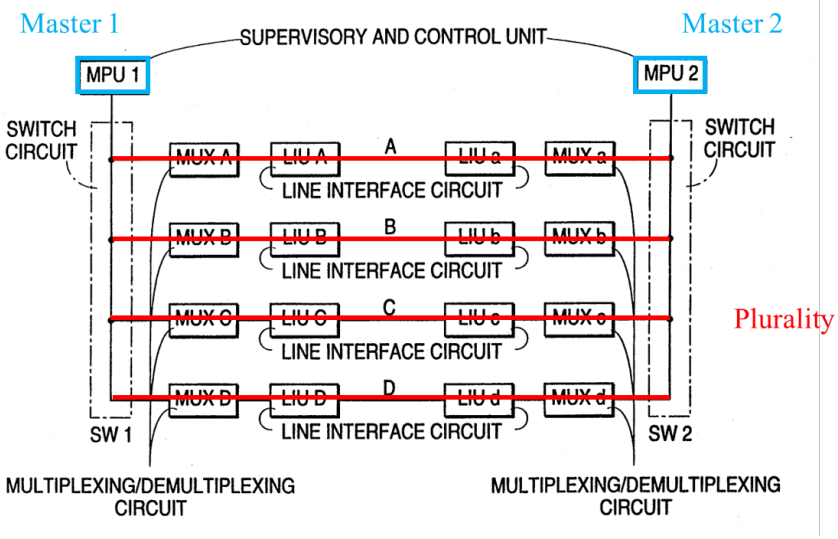
| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | which is valuable for buffer management applications. Moreover, the point-to-point ring topology bus employs a variable latency access method that enables messages to be passed across the bus with low latency when the system is quiet and with increased latency when the system is busy. In addition, since control messaging around the ring interconnect and across the channel interconnects are embedded in the data stream, the number of pins required and manufacturing costs are reduced." Petersen, 20:38-57. |
| 19[e] | in the event of a fault in the daisy chain, conveying further downstream data packets, received from the network by one of the master units, along the daisy chain in a second direction, opposite to the first direction, so as to deliver the further packets to the ports of at least some of the slave units. | The Reference discloses in the event of a fault in the daisy chain, conveying further downstream data packets, received from the network by one of the master units, along the daisy chain in a second direction, opposite to the first direction, so as to deliver the further packets to the ports of at least some of the slave units.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotation added)<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link |

| No. | '904 Patent Claim 19 | The Reference |
|---|---|---|
| | | communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

| No. | '904 Patent Claim 20 | The Reference |
|---|---|---|
| 20 | A method according to claim 19, wherein the initial and further downstream packets are received from the network by the first master unit, and wherein conveying the further downstream packets in the second direction comprises conveying the further downstream packets from the first master unit to the second master unit, and then conveying the further downstream packets from the second master unit to the daisy chain. | The Reference discloses a method according to claim 19, wherein the initial and further downstream packets are received from the network by the first master unit, and wherein conveying the further downstream packets in the second direction comprises conveying the further downstream packets from the first master unit to the second master unit, and then conveying the further downstream packets from the second master unit to the daisy chain.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 21 | The Reference |
|---|---|---|
| 21 | A method according to claim 20, wherein conveying the further downstream packets from the first master unit to the second master unit comprises linking further slave units in an additional daisy chain between the first and second master units, and conveying the further downstream packets from the first master unit to the second master unit over the additional daisy chain. | The Reference discloses a method according to claim 20, wherein conveying the further downstream packets from the first master unit to the second master unit comprises linking further slave units in an additional daisy chain between the first and second master units, and conveying the further downstream packets from the first master unit to the second master unit over the additional daisy chain.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Below are examples of such references.<br><br>**Sugawara discloses**:<br>Sugawara, 3:6-14 ("FIG. 2 is a schematic block diagram illustrating the principle of the present invention. As illustrated, according to the present invention, switch circuits SW1 and SW2 are provided which selectively connect supervisory and control units MPU1 and MPU2 to corresponding ones of multiplexing/demultiplexing circuits MUXA to MUXD and MUXa to MUXd. A backup line P is not provided."). |

| No. | '904 Patent Claim 21 | The Reference |
|---|---|---|
| | | FIG. 2<br>  <br><br>FIG. 2 (annotated).<br><br>Sugawara, 3:24-50 ("When a failure of the line A is detected, the data link communication is switched from the line A to another, for example, the line B. This switching is made as follows. That is, a line failure is usually detected as a failure in reception. When detecting the line failure in terms of error rate by way of example, the line interface circuit LIUa on the receiving side informs the supervisory and control unit MPU2 via MUXa of the detection of the line failure. Upon being informed of the line failure, MPU2 informs MPU1 of the failure of the line A via another line, for example, another line of B directed from MPU2 to MPUl not shown. Responsive to this, the supervisory and control unit MPUl switches switch circuit SWl to connect MPUl to another line, for example, the line B. Thereby, the data link communication becomes effected over the line B.<br><br>If a failure should also occur in the line B, the line switching will be made likewise. The line B is thus switched to another line, for example, the line C. In FIG. 1, if the line A is faulty, it is switched to the backup line P and, if the backup line P is also faulty, the data link |

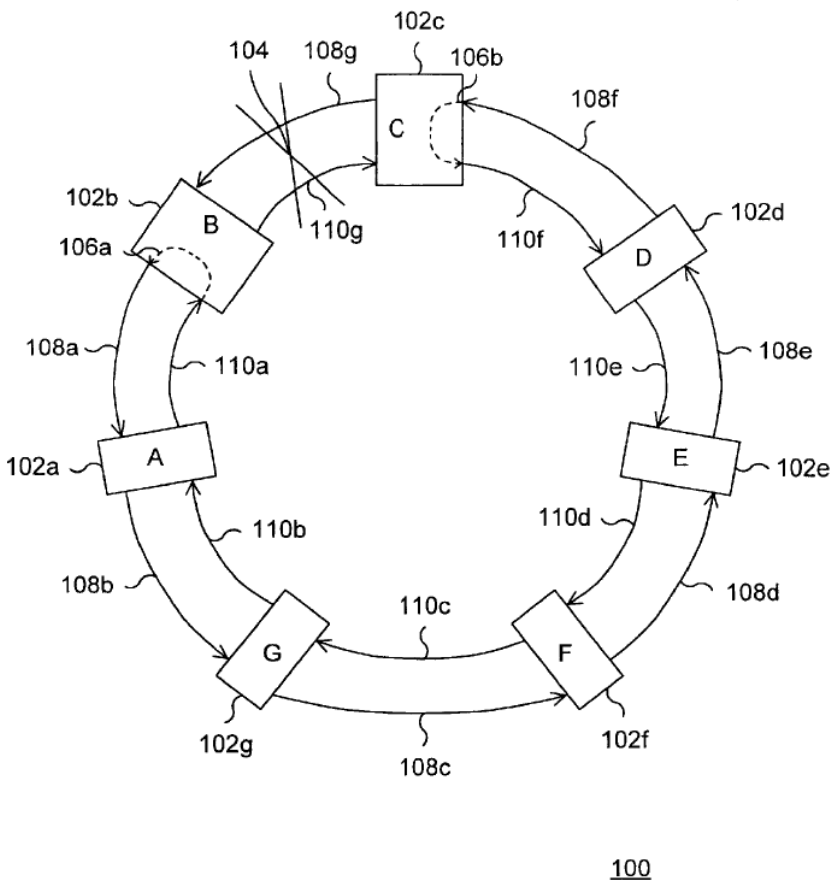
| No. | '904 Patent Claim 21 | The Reference |
|---|---|---|
| | | communication will be interrupted. According to the present invention, as long as there are normal lines, data link communications can be continued. No backup is needed. However, if a failure occurs in a line, the transmission of a main signal over the line is interrupted.").<br><br>Sugawara, Abstract ("A line interface circuit on the receiving side detects communication failure in terms of an error rate and informs the supervisory and control unit on the receiving side of the occurrence of the failure. The supervisory and control unit on the receiving side in turn switches the line data communications from the receiving side to the transmitting side to another line unused for data communications to inform the supervisory and control unit on the transmitting side of the occurrence of the failure. The supervisory and control unit on the transmitting side switches the line for data communications from the transmitting side to the receiving side to the other line. The switching circuit on the receiving side comprises buffers each inserted in a line and a buffer connected to the common outputs of the buffers so that the supervisory and control unit on the receiving side automatically receives a transmit signal transmitted over a line switched on the transmitting side.") |

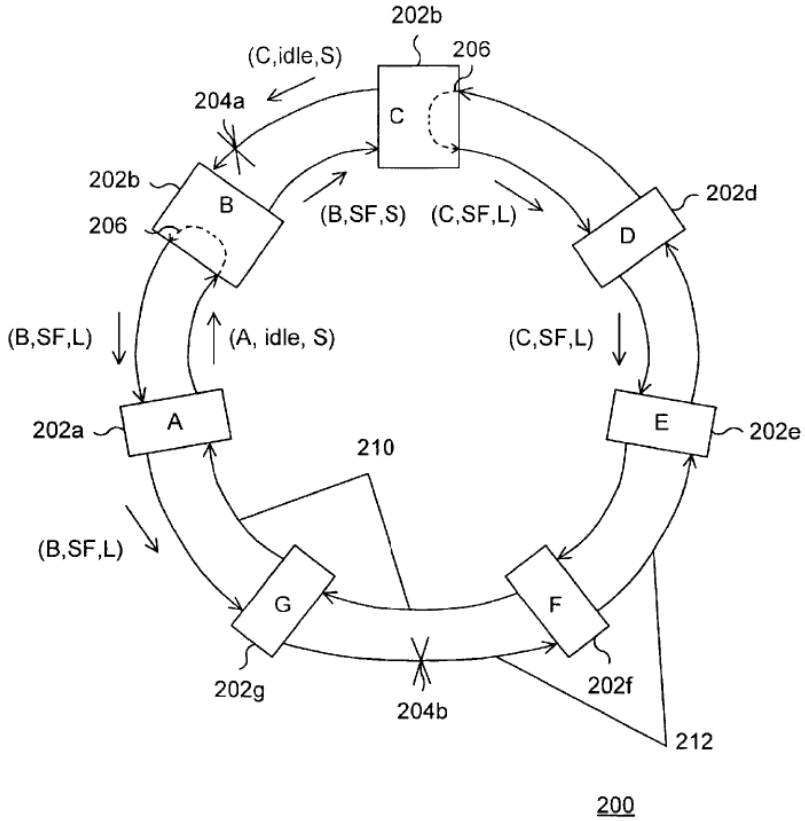
| No. | '904 Patent Claim 22 | The Reference |
|---|---|---|
| 22[a] | A method according to claim 19, and comprising conveying initial upstream data packets, received by the slave units from the subscriber lines, along the daisy chain in the second direction so as to transmit the upstream data packets via the first master unit over the network, | The Reference discloses a method according to claim 19, and comprising conveying initial upstream data packets, received by the slave units from the subscriber lines, along the daisy chain in the second direction so as to transmit the upstream data packets via the first master unit over the network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

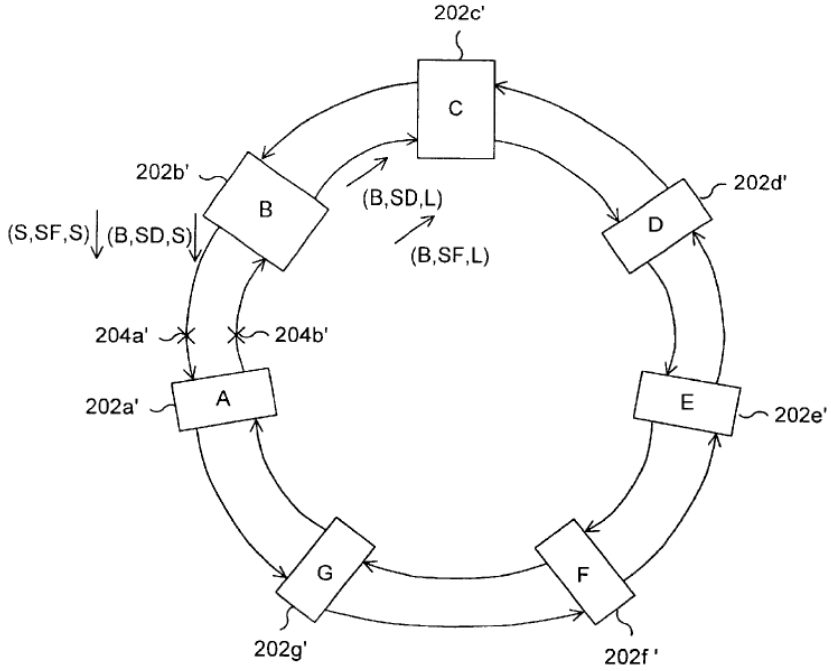
| No. | '904 Patent Claim 22 | The Reference |
|---|---|---|
| 22[b] | and in the event of the fault, conveying further upstream data packets received by one or more of the slave units along the daisy chain in the first direction via the second master unit. | The Reference discloses and in the event of the fault, conveying further upstream data packets received by one or more of the slave units along the daisy chain in the first direction via the second master unit.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 23 | The Reference |
|---|---|---|
| 23 | A method according to claim 22, and comprising bicasting the upstream data packets from the first master unit to the network and to the second master unit, which transmits the bicast upstream data packets over the network. | The Reference discloses a method according to claim 22, and comprising bicasting the upstream data packets from the first master unit to the network and to the second master unit, which transmits the bicast upstream data packets over the network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| 24 | A method according to claim 19, wherein conveying the initial downstream data packets along the daisy chain comprises pre-switching the packets at each of the slave units, so that packets not addressed to any of the ports on the slave unit are passed to the next slave unit in the daisy chain, while packets that are addressed to one or more of the ports on the slave unit are passed to a switch fabric that directs the packets to the ports to which they are addressed. | The Reference discloses a method according to claim 19, wherein conveying the initial downstream data packets along the daisy chain comprises pre-switching the packets at each of the slave units, so that packets not addressed to any of the ports on the slave unit are passed to the next slave unit in the daisy chain, while packets that are addressed to one or more of the ports on the slave unit are passed to a switch fabric that directs the packets to the ports to which they are addressed.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>Cisco commercialized and patented technology relating to monitoring, detecting, and resolving faults without requiring a network reconfiguration *before* Orckit. Some examples of Cisco's patents (and other disclosures) for that technology that are relevant to this limitation include:<br>• Daruwalla<br>• Nederveen<br>• Slater '421<br>• Petersen<br><br>**Daruwalla discloses**:<br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. The present invention provides a protection protocol which simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | independently operating ring networks can be merged and the protection protocol will appropriately remove a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, would remove protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, Abstract.<br>  <br><br>Daruwalla, FIG. 1. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 2. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | |  <br>Daruwalla, FIG. 11.<br><br>"The present invention relates to computer networks. In particular, the present invention relates to a system and method for providing a protection protocol for fault recovery for a two line bi-directional ring network." Daruwalla, 1:8-11.<br><br>"FIG. 1 shows an example of a two line bi-directional ring network. The ring network 100 is shown to include nodes 102 a-102 g. Each node is typically a computer with embedded processors and at least one network connection. Each node 102 a-102 g is shown to be bidirectionally coupled to two neighboring nodes 102 a-102 g via an inner connection ring 110 a-110 g and an outer connection ring 108 a-108 g. For instance, node 102 a is bidirectionally coupled to nodes 102 b and 102 g. The example of FIG. 1 also shows a problem |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | 104 in the connection between node 102 b and node 102 c. When a problem is detected (such as a bi-directional line cut), the connection between nodes 102 b and 102 d wraps back upon itself, as shown by wraps 106 a and 106 b. In this manner, the connection problem 104 can be avoided." Daruwalla, 1:17-30.<br><br>"In a conventional SONET network, each message sent by a sending node to a receiving node typically needs the identification and location of the receiving node to arrive at the proper destination. Accordingly, manual configuration is typically needed in each node to store the identity and location of each other node in the ring network in order to provide for communication between the nodes in the network." Daruwalla, 1:31-44.<br><br>"In summary, for the protection mechanism to operate, each node needs to know the current ring map (current ring topology). What is needed is a system and method for providing fault recovery for two line bi-directional ring network that minimizes the need to keep track of other nodes in the ring network. Preferably, the system would not require reconfiguration of an internal map of the network when a new node is added to, or existing nodes are removed from the network. The present invention addresses such a need." Daruwalla, 2:23-31.<br><br>"The present invention provides a protection protocol for fault recovery, such as a ring wrap, for a network, such as a two line bi-directional ring network. An embodiment of the present invention works in conjunction with a ring topology network in which a node in the network can identify a problem with a connection between the node and a first neighbor. According to the embodiment, when the problem is identified, the node sends a message identifying the problem to a second neighbor which is located at least one node away from the problem. The second neighbor then forwards the message to a third neighbor, unless the second neighbor is dealing with a situation that is higher in a hierarchy of situations than the problem described in the message by the original node. In general, if the second neighbor's situation has a higher priority than the situation described by the original node, then the message is ignored and not forwarded. If, however, the message sent by the original node describes a situation with a higher priority than the situation being dealt with by the second neighbor, then, in general, the second neighbor's situation is ignored, at least for the moment, and the original node's message is forwarded to the next neighbor. In general, a higher priority request preempts a |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | lower priority request within the ring. Exceptions are noted as rules of the protection protocol." Daruwalla, 2:35-58.<br><br>"The present invention provides a protection protocol that simplifies the coordination required by the nodes in a ring network. The nodes do not need to maintain a topology map of the ring, identifying and locating each node on the ring, for effective protection. Additionally, independently operating ring networks can be merged and the protection protocol automatically appropriately removes a protection, such as a ring wrap, to allow the formation of a single ring. It also provides for multiple levels of protection priority so that protection for a high priority failure, such as a physical break in a connection, removes protection for a low priority failure, such as a signal degrade, on another link." Daruwalla, 2:59-3:3.<br><br>"A method according to an embodiment of the present invention for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:4-12.<br><br>"In another aspect of an embodiment of the present invention, a method for adding a new node to a ring computer network, the ring network including a plurality of nodes, is presented. The method comprises detecting a situation by a first node, wherein the first node is one of the plurality of nodes; sending a first message via a short path to a second node, wherein the first node is adjacent to the second node prior to an addition of the new node; initiating a fault recovery procedure when the second node receives the first message; receiving a second message from the new node; and entering an idle state when the second message is received." Daruwalla, 3:13-24. |

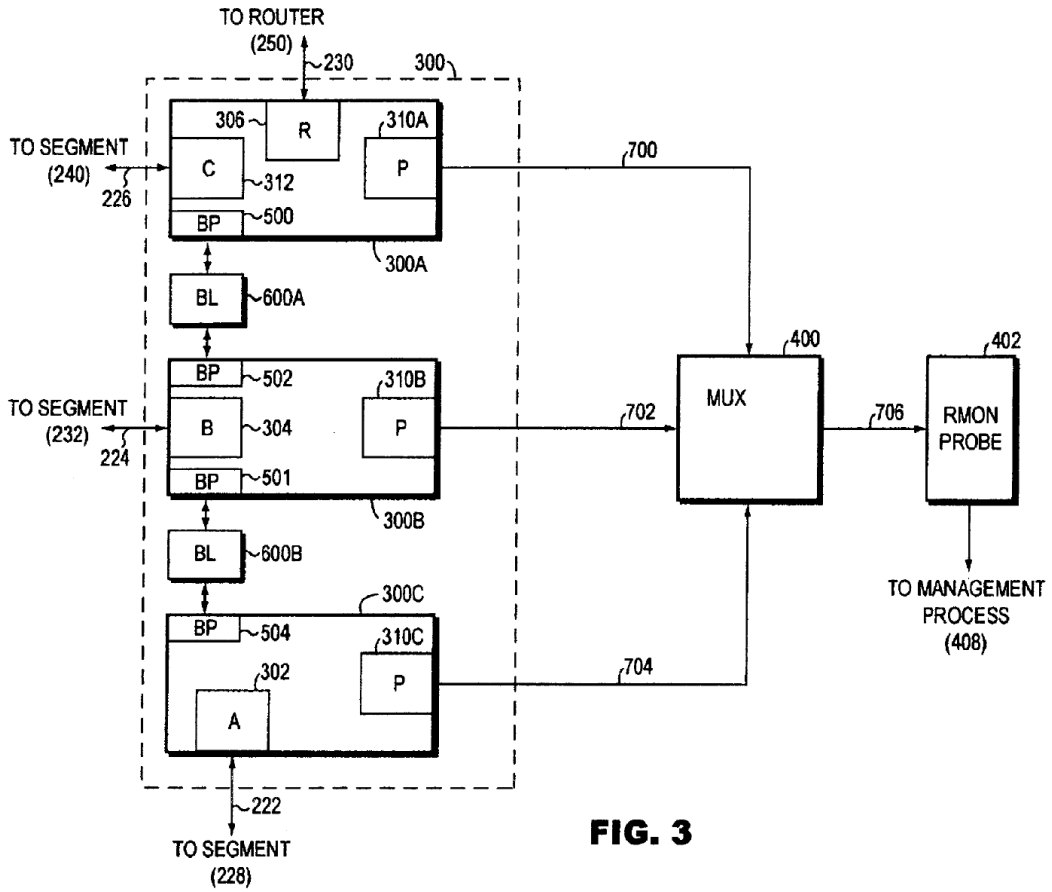
| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "In yet another aspect of an embodiment of the present invention, a system for fault recovery for a ring computer network, the ring network including a plurality of nodes, is presented. The system comprises means for detecting a situation by a first node, wherein the first node is one of the plurality of nodes; means for sending a first message via a short path to a second node, wherein the first node is adjacent to the second node; and means for initiating a fault recovery procedure when the second node receives the first message." Daruwalla, 3:25-35.<br><br>"FIG. 2 is block diagram of a ring network utilizing a protection protocol according to an embodiment of the present invention." Daruwalla, 3:40-42.<br><br>"FIGS. 4-6 are flow diagrams illustrating various rules within the protection protocol according to an embodiment of the present invention." Daruwalla, 3:46-48.<br><br>"FIGS. 8-12 are flow diagrams and a system diagram illustrating further rules of the protection protocol according to an embodiment of the present invention." Daruwalla, 3:52-54.<br><br>"FIG. 2 is a block diagram showing a ring network system utilizing a method of fault recovery according to an embodiment of the present invention. The ring network 200 is shown to include nodes 202 a-202 g. The nodes 202 a-202 g are shown to be coupled via an inner ring 210 in which the data flows in one direction, such as a clockwise direction. Additionally, the nodes 202 a-202 g are also shown to be coupled by an outer ring 212 in which data can flow in the opposite direction to the inner ring 210, such as in a counter-clockwise direction. The ring network 200 is shown to have a situation 204 a that requires protection, such as a ring wrap 206." Daruwalla, 5:35-45.<br><br>"FIG. 4 is a flow diagram of an example of a method according to an embodiment of the present invention implied by Rules 1-22. An APS packet is received via step 400. It is determined whether the APS packet has been sent along a long path via step 402. If the packet was not sent via a long path, then the APS packet is not forwarded via step 406. Accordingly, if the APS packet was sent via the short path, then the packet is not forwarded via step 406. If, however, the packet was sent through the long path via step 402, then the APS packet may |

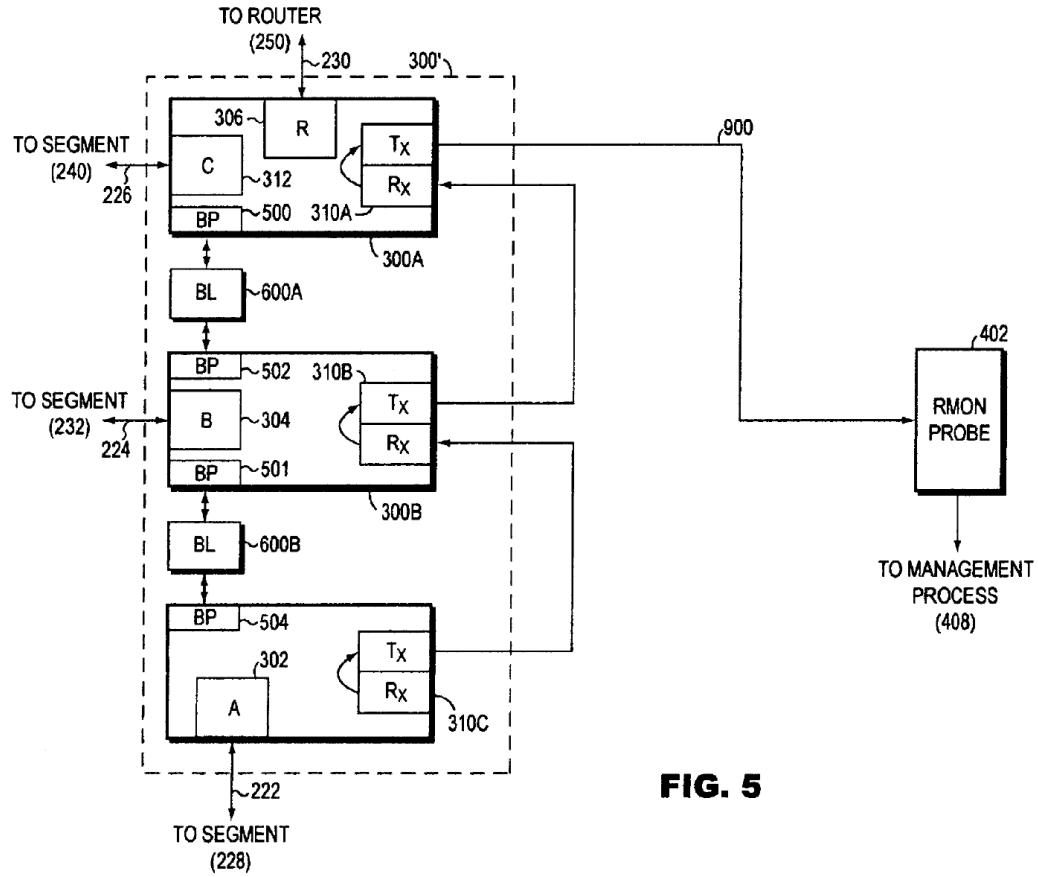
| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | be forwarded via step 404. Note that for this example of Rule (1), it is assumed that the long path message does not have to pass through a wrapped connection in order to be forwarded. Otherwise, if the long path message needs to pass through a wrapped connection in order to be forwarded, then the message will simply not be forwarded." Daruwalla, 6:21-36.<br><br>"FIG. 6 is a flow diagram illustrating Rules 4 and 5. A node detects a problem between the node and a first neighbor via step 600. The node performs a wrap away from the side on which the problem exists via step 602. A short path message is then sent to the first neighbor informing it of the problem via step 604. Additionally, a long path message is also sent to a second neighbor informing the second neighbor of the problem via step 604. The first neighbor then performs a wrap away from the side of the problem via step 606. The first neighbor also sends an IDLE message, indicating a wrapped status, on a short path to the node that detected the problem via step 608. This message is sent across the failed span. Note that IDLE messages do not get wrapped and are sent across failed spans if possible. Additionally, the first neighbor also sends a message on a long path toward the side without the problem via step 608." Daruwalla, 6:64-11.<br><br>"An example of the method described in FIG. 6 can be seen in FIG. 2. Node 202 b has detected a problem 204 a and performs a wrap 206 on the side on which the problem exists. Node 202 b also sends a short path message to the neighbor on the other side of the problem 204 a, which is node 202 c. Node 202 b also sends a long path message to its other neighbor node 202 a informing it of the problem. Node 202 c performs a wrap 206 on the side of the problem and sends an IDLE message on a short path to node 202 b. Node 202 c also sent a message on a long path toward the side without the problem to its neighbor 202 d." Daruwalla, 7:12-21.<br><br>"FIG. 7 lists the hierarchy of priorities of Rule (8). For ease of reference, the hierarchy is separated into Class I-III. Class I is the highest priority, while Class III is the lowest priority. An example of a highest priority message in Class I is lockout. Lockout is an order stating that the ring network is not to wrap under any circumstances." Daruwalla, 7:22-26. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "Examples of the next priority listed in Class II are forced switch and signal fail. Forced switch indicates that the ring network is configured to wrap at the point of the forced switch. Signal fail is a situation where either two nodes cannot communicate with each other, or one node cannot hear the other node. An example of a signal fail is a physical break in the communication lines between two nodes." Daruwalla, 7:27-33.<br><br>"Note that members of Class II can co-exist (Rule 9). For example, multiple forced switches and signal fails can co-exist. Additionally, members of Class I can co-exist (Rule 10). For example, multiple lockouts in a single ring network can co-exist. However, situations in Class III cannot co-exist with other situations (Rule 11). For example, a signal degrade cannot co-exist with a wait-to-restore." Daruwalla, 7:52-58.<br><br>"When there are multiple requests of the same priority within Class III, the first request to complete a long path signaling will take priority (Rule 13). For example, if there are two signal degrades located on the same ring network, then the first signal degrade which completes the long path signaling will take priority over the other signal degrade. By not allowing members of Class III to co-exist with one another, partitioning of the ring network is avoided." Daruwalla, 7:59-65.<br><br>"In case of two equal requests within Class III on both inner and outer rings of the ring network, the tie is broken by choosing a request on one of the rings, such as the outer ring request (Rule 14). For example, if a signal degrade occurs both on the inner and outer rings, then a request on a predetermined ring, such as the outer ring, will take priority over the other requests." Daruwalla, 7:66-8:5.<br><br>"FIG. 8 is a flow diagram illustrating Rules (9), (10), (11), (13), and (15). Note that the flow diagram described in FIG. 8 is merely an example of one way in which the rules of the method according to the embodiment of the present invention can be executed. For example, the determination of whether the long path message is a Class I request via step 802 or a Class II request via step 810 can be in reverse order." Daruwalla, 8:6-11. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "A wrapped node receives a long path message via step 800. It is then determined if the long path message is a Class I request via step 802. The classes used in FIG. 8 are meant to correspond with the example of classes defined in FIG. 7. If the long path message is a Class I request, then it is determined if a local situation also has a Class I request via step 804. A local situation includes scenarios such as when a node detects a situation or problem, or when a node is made aware of a problem or situation via a short path message from its neighbor. If a local situation is not a Class I request via step 804, then any existing local wraps are unwrapped and the long path message is forwarded via step 806. If, however, a local situation is a Class I request via step 804, then the connections are already unwrapped or was never wrapped, and the long path message is forwarded via step 808." Daruwalla, 8:12-26.<br><br>"FIG. 12 is a flow diagram illustrating rules (20) and (21) of the method according to the embodiment of the present invention. A wrapped node determines that a problem has been cleared via step 1200. It then enters a wait-to-restore state via step 1202. It is then determined if its neighbor is the same neighbor as previously noted via step 1204. The node can save the source of a short path message at the time of wrap initiation to note the identity of its neighbor. If the current neighbor is not the same as the previous neighbor via step 1204, then an IDLE state is entered via step 1206. If, however, the current neighbor is the same as the previous neighbor via step 1204, then it is determined whether a pre-determined wait-to-restore time has expired via step 1208. Once the pre-determined wait-to-restore time has expired, then the node enters an IDLE state via step 1206." Daruwalla, 12:60-13:6.<br><br>"A method and system for fault recovery for a two line bi-directional network has been disclosed. Software written according to the present invention may be stored in some form of computer-readable medium, such as memory or CD-ROM, or transmitted over a network, and executed by a processor." Daruwalla, 13:7-19. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | **Nederveen discloses**:<br>"A technique for use in gathering network activity-related information from cascaded network switches is provided. Using this technique, the information can be gathered without substantially reducing performance of the cascaded switches. In one embodiment, a single remote monitoring probe is connected via respective connections to each of the switches so as to receive the information from the switches. In another embodiment, only one of the switches is connected to the probe, and the other switches transmit their respective portions of the information to the switch connected to probe. The switch connected to the probe provides these portions of the information, as well as, any of its respective activity-related information to the probe. In this latter embodiment, the switches may be connected by dedicated connections and switch ports that are used solely for communicating the activity-related information." Nederveen, Abstract. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | |  <br>**FIG. 3**<br>Nederveen, FIG. 3. |

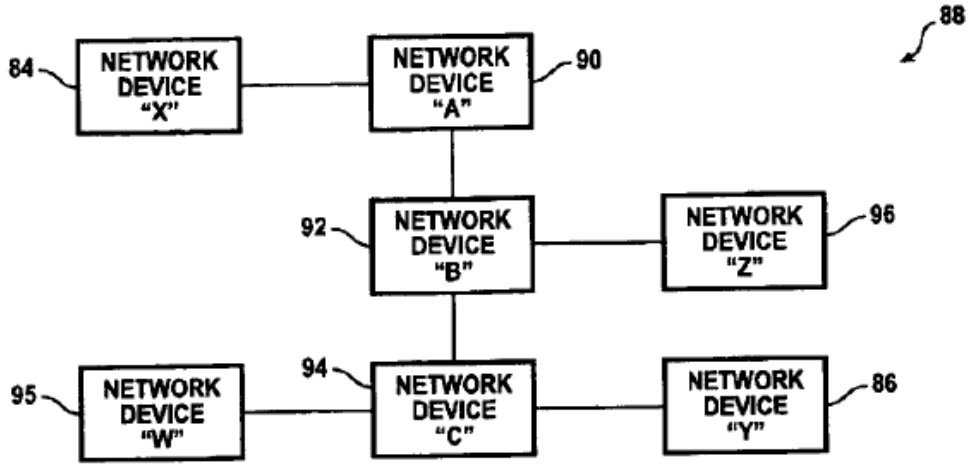
| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | |  <br>**FIG. 5**<br><br>Nederveen, FIG. 5.<br><br>"Thus, it would be desirable to provide a stacked switch monitoring technique that permits efficient offloading of raw data processing from the stacked switches, requires only a minimal number of specialized network entities to gather and process such raw data, and does not result in substantial degradation of stacked switch performance." Nederveen, 4:38-43. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "Accordingly, the present invention provides a technique for remote monitoring of a switch network that overcomes the aforesaid and other disadvantages and drawbacks of the prior art. More specifically, in one aspect of the present invention, a technique is provided for gathering information that may be useful in network management (e.g., switch port activity-related information), from switches in the network that are in a stacked configuration. The information is gathered from the stacked switches by a single network entity (e.g., an SNMP remote monitoring probe) in such a way that it does not substantially degrade the performance of the switches. This is accomplished, in one embodiment of the technique of the present invention, by connecting the switches via respective connections to a multiplexer that selectively connects the switches, according to an arbitration scheme, to the single network entity. The entity gathers respective portions of the information from switches when it is connected to the switches by the multiplexer. The information gathered by the entity may be provided to another network entity (e.g., an SNMP management node) in order to permit the other entity to use that information in managing the network." Nederveen, 4:46-67.<br><br>"In another embodiment of the technique of the present invention, only one of the switches is connected to the single information gathering entity. The switches that are not connected to the entity transmit, via respective dedicated ports and connections (i.e., ports and connections that are used solely for network information gathering activities), their respective portions of the information to the switch that is connected to the entity. The switch that is connected to the entity transmits, via a respective dedicated port and connection, the information received from the other switches, as well as, its own information to the entity." Nederveen, 5:1-11.<br><br>"FIG. 3 is a schematic, functional block diagram illustrating in greater detail the construction of the stacked switch network shown in FIG. 2." Nederveen, 5:26-28.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network configured to employ another embodiment of the present invention." Nederveen, 5:32-34. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "FIGS. 2-5 illustrate features of a computer network 200 wherein embodiments of the present invention may be advantageously practiced. Network 200 comprises a stacked switch network 300 which interconnects a plurality of network segments 228, 232, 240, and 251. Each segment 228, 232, 240 comprises one or more local area networks having computer endstations (not shown). Segment 251 is a network router segment that comprises network router 250. Each segment 228, 232, 240 is coupled via a respective communications link 222, 224, 226 to a respective port 302 (i.e., port A), 304 (i.e., port B), 312 (i.e., port C) of the switch network 300. Likewise, the router 250 of router segment 251 is coupled via a respective trunk line 230 to router port 306 (i.e., port R)." Nederveen, 5:46-59.<br><br>"Stacked switch network 300 comprises a plurality of data network switches 300A, 300B, 300C (e.g., Catalyst 3900™ series switches of the type commercially available from the Assignee of the subject application) coupled together via conventional stack link bus connection logic 600A, 600B. More specifically, logic 600A couples a stack link bus port and associated logic 500 in switch 300A to a stack link bus link port and associated logic 502 in switch 300B. Similarly, logic 600B couples another stack link bus port and associated logic 501 in switch 300B to a stack link bus port and associated logic 504 in switch 300C. It should be understood that although, as is shown in FIG. 3, switches 300A and 300B, and switches 300B and 300C, may be coupled serially together by separate respective logic elements 600A, 600B, each of the switches 300A, 300B, 300C may be coupled together via a single respective stack link bus port in the switch to a single stack link bus connection logic block (not shown, e.g., of the type that is commercially available under the tradename Catalyst Matrix™ from the Assignee of the subject application). Further alternatively, depending upon the particular design and functionality of the ports 500, 501, 502, and 504, and the control and forwarding logic (whose operation will be described more fully below) in the switches 300A, 300B, 300C, the circuitry in logic 600A, 600B may instead be comprised in the ports 500, 501, 502, and 504 and/or control and forwarding logic, and therefore, in this alternative configuration, the logic 600A, 600B in the network 300 may be replaced by simple connection means (e.g., cable connectors)." Nederveen, 6:29-57. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "Each switch 300A, 300B, 300C includes a respective internal bus (e.g., element 800 in switch 300C) that is coupled via at least one stack link bus port and associated interface logic (e.g., 504 in switch 300C) to external stack link bus connection logic (e.g., element 600B in switch 300C). Each switch 300A, 300B, 300C also includes respective programmable control and forwarding logic (e.g., element 802 in switch 300C) comprising processing, memory, and other circuitry for storing and learning configuration information (e.g., source and destination MAC addresses of messages received by the switch, switch bridging table, switch segments' spanning tree and virtual local area network information, etc.), and for providing appropriate commands to other elements (e.g., the switch ports) to cause data messages received by the switch to be forwarded to appropriate network segments coupled to the switch based upon this configuration information. In each switch, the switch's port logic circuitry (e.g., port A logic 302 and port P logic 310C in switch 300C) and control and forwarding logic are coupled to each other via that switch's respective internal bus. The stack link bus port and associated logic in each switch 300A, 300B, 300C may comprise a Catalyst™ stack port line interface card (commercially available from the Assignee of the subject application) inserted into a bus expansion slot (not shown) in the switch. Although not shown in the Figures for purposes of clarity of illustration, each switch 300A, 300B, 300C in network 300 typically will include tens or hundreds of ports coupled to network segments." Nederveen, 6:58-7:19.<br><br>"The control and forwarding logic and stack link bus port and associated logic in each switch, and the logic 600A, 600B, are configured to together implement conventional techniques for permitting the switches 300A, 300B, 300C to function together as a single logical/virtual switch. More specifically, when configured in the stacked arrangement 300, after the switches 300A, 300B, 300C and logic 600A, 600B are initially activated, they execute initial power-on self-diagnostics, and thereafter, enter a "stack discovery" mode of operation." Nederveen, 7:20-29.<br><br>"In the stack discovery mode of operation, the control and forwarding logic in each switch 300A, 300B, 300C first "senses" that its switch is coupled to logic 600A and/or 600B, and then determines the particular configuration of the stacked switch network 300, using suitable conventional autosensing/autoconfiguration techniques. The control and 353orwarding logic in the switches 300A, 300B, 300C then assigns to the switches respective unique |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | identification numbers (e.g., based upon unique identification numbers of respective ports of the logic 600A, 600B to which the switches are coupled)." Nederveen, 7:30-40.<br><br>"FIG. 5 is schematic, functional block diagram illustrating construction of a stacked switch network 300′ configured to employ another embodiment of the present invention. It should be understood that unless specifically stated to the contrary, the structure and operation of the network 300′ are substantially the same as the structure and operation of network 300. In network 300′, each of the dedicated ports 310A, 310B, 310C comprises a respective transmit portion and receive portion, referenced in FIG. 5 as RX and TX, respectively." Nederveen, 11:20-29.<br><br>**Slater '421 discloses**:<br>"A method and apparatus for discovering paths to other network devices includes a protocol and network management application that can be executed on network devices. The Ethernet protocol is used to detects paths to other network devices, knowing only the Ethernet address of the destination. A discovery protocol is extended to add hop probe and hop probe reply Type-Length-Value fields in a variable-length list. The hop probe fields contain a hop count, a destination Ethernet address, and a source Ethernet address. When a hop probe is received by a network device, the hop count field is decremented by one and the hop probe is forwarded. Packet received with a hop count of one are not forwarded and a hop probe reply is sent back to the Ethernet source address of the hop probe. The hop probe reply fields contain a destination Etherned address and a source Ethernet address." Slater '421, Abstract. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br><br>Slater '421, FIG. 6.<br><br>"Partly as a result of the increased complexity of networks, network administrators must often troubleshoot problems with their network. Two classes of network problems often faced by network administrators are "reachability" problems and performance slowdowns. Reachability problems occur when one or more network devices cannot be accessed through a network, and can be caused by hardware or software failures, cabling problems, or any of several other types of difficult-to-diagnose problems that can occur in a network." Slater '421, 7:11-20.<br><br>"Some of the world's largest networks today rely on the TCP/IP suite of networking protocols. With a relatively small kit of basic tools, network administrators can learn much about an internetwork. "Ping" and "traceroute" commands, "show" commands, and "debug" commands (all of which are typically available via the basic management interface on a network device) form the core of the network administrator's internetwork toolkit. Ping and |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | traceroute commands can be useful tools in determining where failures are occurring, but they are cumbersome to use, and require knowledge of the IP address or host name of the destination network device. The show commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data. Finally, debug commands provide clues about the status of a network device and other network devices directly or indirectly connected to it. Because debug commands can create performance slowdowns, they must be used with great care, and using the wrong debug command at the wrong time can exacerbate problems in already poorly performing networks." Slater '421, 7:55-8:8.<br><br>"Embodiments of the present invention as illustrated herein use the Cisco™ Discovery Protocol ("CDP") to automatically detect paths to specified network devices in Ethernet LANs. However, other similar products known to those of ordinary skill in the art are available from other vendors to accomplish the same task." Slater '421, 9:10-15.<br><br>"CDP is a media-independent device discovery protocol which can be used by a network administrator to view information about other network devices directly attached to a particular network device. In addition, network management applications can retrieve the device type and SNMP-agent address of neighboring network devices. This enables applications to send SNMP queries to neighboring devices. CDP thus allows network management applications to discover devices that are neighbors of already known devices, such as neighbors running lower-layer, transparent protocols." Slater '421, 9:16-26.<br><br>"It is to be understood that the present invention is not limited to devices that are compatible with CDP. CDP runs on all media that support the Subnetwork Access Protocol ("SNAP"), including LAN and Frame Relay. CDP runs over the data link layer only. Each network device sends periodic messages to a multicast address and listens to the periodic messages sent by others in order to learn about neighboring devices and determine when their interfaces to the media go up or down. Each device also advertises at least one address at which it can receive SNMP messages. CDP messages, or "advertisements," contain holdtime information, which indicates the period of time a receiving device should hold CDP information from a neighbor before discarding it. With CDP, network management applications can learn the device type |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | and the SNMP-agent address of neighboring devices. This process enables applications to send SNMP queries to neighboring devices." Slater '421, 9:27-43.<br><br>"It should be noted that-normally, CDP packets according to aspects of the present invention are transmitted at regular intervals (e.g. once every 60 seconds). However, in embodiments of the present invention, when a Hop Probe or Hop Probe Reply needs to be forwarded by a network device, the network device is commanded to send a CDP packet immediately." Slater '421, 16:66-17:5.<br><br>"The present invention is much faster than the previous method that involved logging in to each intermediate network device, entering the "show cdp neighbors" command, and interpreting the output to find the next hop along the path to the destination network device. Also, the present invention allows individual users, such as network administrators, to execute a tool to manually discover paths through a network of Ethernet switches. The present invention can be used by network management software to automatically map the topology of clusters of network devices, such as Ethernet switches. Finally, the present invention is useful in loop detection. Enhancements to Spanning Tree and other bridge-level routing protocols can test proposed changes to switch topology prior to making them." Slater '421, 17:6-20.<br><br>**Petersen discloses**:<br>"Methods and apparatus for enabling communication between a source network device and one or more destination network devices are disclosed. A system enabling communication between a source network device and one or more destination network devices includes a switch and a ring interconnect. The switch is adapted for connecting to the source network device and the one or more destination network devices. More particularly, the switch is capable of storing data provided by the source network device and retrieving the data for the one or more destination network devices. The ring interconnect is adapted for connecting the source network device and the one or more destination network devices to one another. In addition, the ring interconnect is capable of passing one or more free slot symbols along the ring interconnect. Thus, the ring interconnect is capable of expanding when one or more of the free slot symbols are each replaced by a frame notify message indicating that the data has |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | been stored by the switch for retrieval by the one or more destination network devices." Petersen, Abstract. |

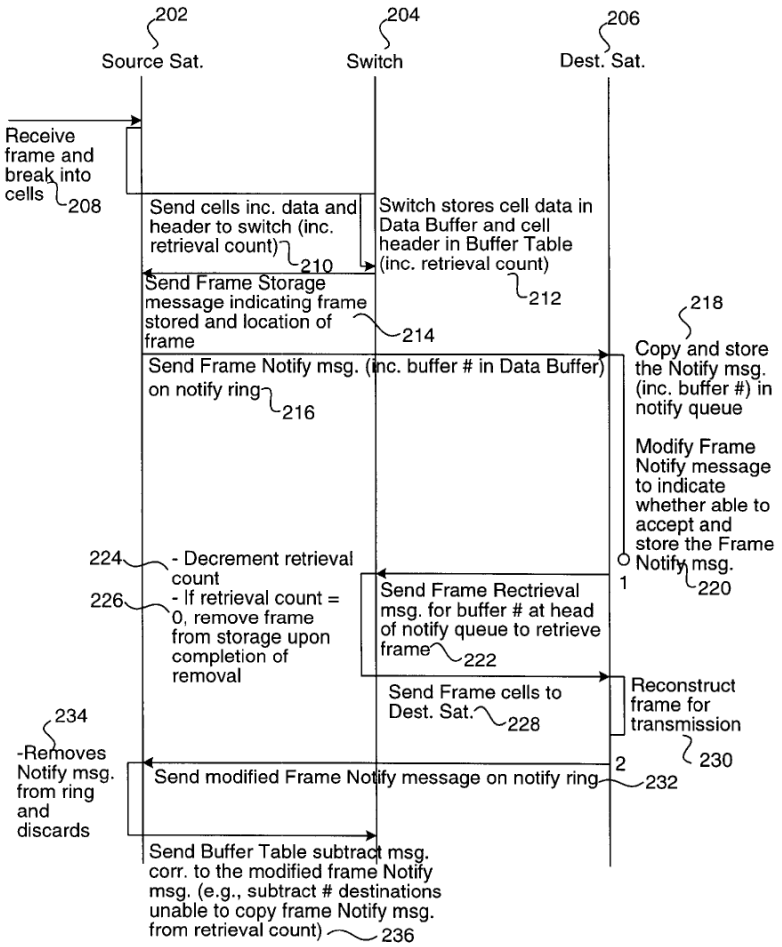



***FIG. 2***

Petersen, FIG. 2.

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "The present invention relates to a mixed topology data switching system that combines a radial interconnect with a ring interconnect. More particularly, the radial interconnect permits devices to store and retrieve data using a switch, while the ring interconnect permits devices along the ring interconnect to provide notification that data has been stored for retrieval, as well as provide feedback regarding the ability or inability to retrieve such data." Petersen, 1:34-41.<br><br>"In controlling the flow of network traffic through a switching system, it is often desirable to provide feedback to the source of the data. For instance, although a transmitting device, hereinafter referred to as a "source device," may transmit or forward data to a receiving device, hereinafter referred to as a "destination device," the destination device may be incapable of handling the data. In these circumstances, the source device is often unaware that the data was not accepted by the destination device, complicating switch management. Common solutions to the problem of switch traffic management have included ensuring that all intended destination devices are "ready to receive" prior to transmitting data on a ring or bus interconnect, or insisting that each intended destination device send an explicit acknowledgement back to the source device. Both of these approaches result in reduced efficiency of the interconnect scheme. By way of example, in a ring network, such acknowledgment is typically provided in the data frame being transmitted. As another example, in other interconnect schemes, each such device may send a separate acknowledgment, therefore adding to the traffic on the network. Accordingly, it would be desirable if a traffic management scheme were established which could provide such feedback to the source of the data while minimizing traffic management overhead." Petersen, 2:8-32.<br><br>"According to one embodiment, the present invention combines the use of two data transport methods: a point-to-point radial interconnect and a ring interconnect. The radial interconnect connects interface devices to each other through the services of a central switch device to permit the transport of data. Typically, a single interface has a single dedicated radial interconnect to the central switch. These interface devices are further connected to one another via a ring interconnect to convey retrieval notifications regarding forwarding of the data (by source devices) and receipt of the data (by destination devices)." Petersen, 2:36-46. |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | "Each radial interconnect provides a narrow, high baud-rate connection to convey to the actual data from and to the associated interface without being burdened by the unrelated traffic for the remaining interfaces in the system. This is accomplished through the use of a central switch device, which stores and retrieves data for the various interfaces in the system. As will be apparent from the following description, this architecture provides numerous advantages over a wide parallel bus or ring." Petersen, 2:47-55.<br><br>"The ring interconnect may be used to convey a "retrieval notification"(i.e., retrieval message) that may be observed by all potential retrieving interfaces. The retrieval notification notifies specific devices ("destination devices") or interfaces that one or more frames addressed to them are available from the switch device. Moreover, the ring interconnect permits each destination device to provide feedback to the source device letting the source know whether the destination has accepted the notification provided by the source device and therefore whether the destination can retrieve the data intended for it. The feedback is particularly useful in buffer management applications. In this manner, an efficient and flexible data transport and retrieval notification system that includes a feedback path to the source of the data is provided." Petersen, 2:56-3:3.<br><br>"FIG. 2 is a process flow diagram illustrating a method of providing network communication according to an embodiment of the invention." Petersen, 3:9-11.<br><br>"FIG. 2 is a process flow diagram illustrating in further detail a method of providing network communication in the above-described system according to an embodiment of the invention. As shown, process steps performed by a source device 202 are illustrated along an associated vertical line, steps performed by a switch 204 are illustrated along another vertical line, and steps performed by a destination device 206 are illustrated along still another vertical line." Petersen, 4:50-57.<br><br>"When the frame is stored by the switch 418, the source device preferably receives an acknowledgment that the data has been stored. Thus, to provide this feedback, a frame storage message (i.e., storage reply) is sent from the switch 418 on the channel 416 to the channel interface 414. The frame storage message is then provided to the notify ring interface as |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | shown at 430 and sent on the notify ring. Once this acknowledgment is received by the interface device 402, the designated destination devices may be notified via notify ring interface 424. As described above, a Frame Notify message may be sent via the notify ring interface 424 to the destination devices. More particularly, the Frame Notify message may identify one or more destination devices for the frame and specify the location of the frame to be retrieved. By way of example, the location of the frame to be retrieved by the destination devices may be designated by a buffer number 430. In addition, the destination devices for the frame may be specified in the Frame Notify message through a notify queue map 426. More particularly, the notify queue map 426 may specify a notify queue associated with a particular destination device. The notify queue may be expressly designated through the use of one or more bits as well as implied through the specification of a priority level for the data. The notify queue map 426 will be described in further detail with reference to FIG. 13. The notify ring interface 424 then creates a Frame Notify message including the notify queue map 426 and the buffer number 430 which is then sent on an outbound interface of the notify ring 432." Petersen, 7:55-8:16.<br><br>"As described above, the notify ring may be expanded to accommodate communication between interface devices. The communication between the interface devices and the switch is therefore performed on one or more channels rather than the notify ring. As a result, the flexibility of the notify ring does not effect the speed with which the interface devices may communicate with the switch. Thus, where a single port operates at a faster speed than the channels, multiple channels may be grouped together. In this manner, the speed with which the switch may communicate with the interface devices may be maximized." Petersen, 20:27-36.<br><br>"The present invention provides a mixed topology data switching system that combines a point-to-point radial interconnect with a ring interconnect to maximize the speed of network traffic. The radial interconnect provides a narrow, high baud-rate connection to convey the data traffic for just the interface in question, without being burdened by all of the unrelated traffic for the remaining interfaces in the system. At the same time, the ring interconnect permits retrieval notifications to be observed by all potential retrieving interfaces. The ring topology further permits each destination interface to provide feedback to the source interface, |

| No. | '904 Patent Claim 24 | The Reference |
|---|---|---|
| | | which is valuable for buffer management applications. Moreover, the point-to-point ring topology bus employs a variable latency access method that enables messages to be passed across the bus with low latency when the system is quiet and with increased latency when the system is busy. In addition, since control messaging around the ring interconnect and across the channel interconnects are embedded in the data stream, the number of pins required and manufacturing costs are reduced." Petersen, 20:38-57. |

| No. | '904 Patent Claim 25 | The Reference |
|---|---|---|
| 25 | A method according to claim 19, wherein the network comprises an asynchronous transfer mode (ATM) network. | The Reference discloses a method according to claim 19, wherein the network comprises an asynchronous transfer mode (ATM) network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* Claim 2. |

| No. | '904 Patent Claim 26 | The Reference |
|---|---|---|
| 26 | A method according to claim 19, wherein the network comprises an Internet protocol (IP) network. | The Reference discloses a method according to claim 19, wherein the network comprises an Internet protocol (IP) network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or in view of one or more of the following references: the Catalyst XL Switches, the BayStack 450 Switches, TRENDnet Stackable Hubs, Czerwiec, Quoc, Vink, and Dowling.<br><br>*See supra* Claim 3. |

**EXHIBIT E-2**

Defendant's First Amended Invalidity Contentions
*Orckit Corporation v. Cisco Systems, Inc.*, 2:22-cv-00276-JRG-RSP

---

**Chart for U.S. Patent 8,830,821 ("the '821 Patent")**
**U35 U.S.C. § 103**

In this chart, "The Reference" refers to each of the following references and/or systems:

- Cisco IOS System ("Cisco IOS system")
- Juniper Networks' Junos OS Release 11.1 ("Juniper IOS System")
- MPLS-TP IETF Standards ("IETF MPLS-TP System")
- U.S. Patent Publication No. 2004/0205239 to Doshi ("Doshi '239")
- U.S. Patent No. US 2005/0083928 A1 to Sivabalan, *et al.* ("Sivabalan '928")
- U.S. Patent No. US 2006/0250948 A1 to Zamfir, *et al.* ("Zamfir '948")

The following additional references are identified individually:

- Kurose, James. F. *et al.*, COMPUTER NETWORKING: A TOP-DOWN APPROACH FEATURING THE INTERNET, i-ii, 280-282 (1st ed. 2001) ("Kurose")
- U.S. Patent No. 9,014,049 to Filsfils *et al.* ("Filsfils")
- U.S. Patent No. 8,553,533 to Taylor *et al.* ("Taylor")
- U.S. Patent No. 8,072,879 to Vasseur *et al.* ("Vasseur '879")
- U.S. Patent No. 8,885,460 to Vasseur *et al.* ("Vasseur '460")
- U.S. Patent Application Publication No. 2011/0242988 to Rustogi *et al.* ("Rustogi")

As shown in the chart below, all Asserted Claims of the '821 patent are invalid under 35 U.S.C. § 103 because The Reference renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions.

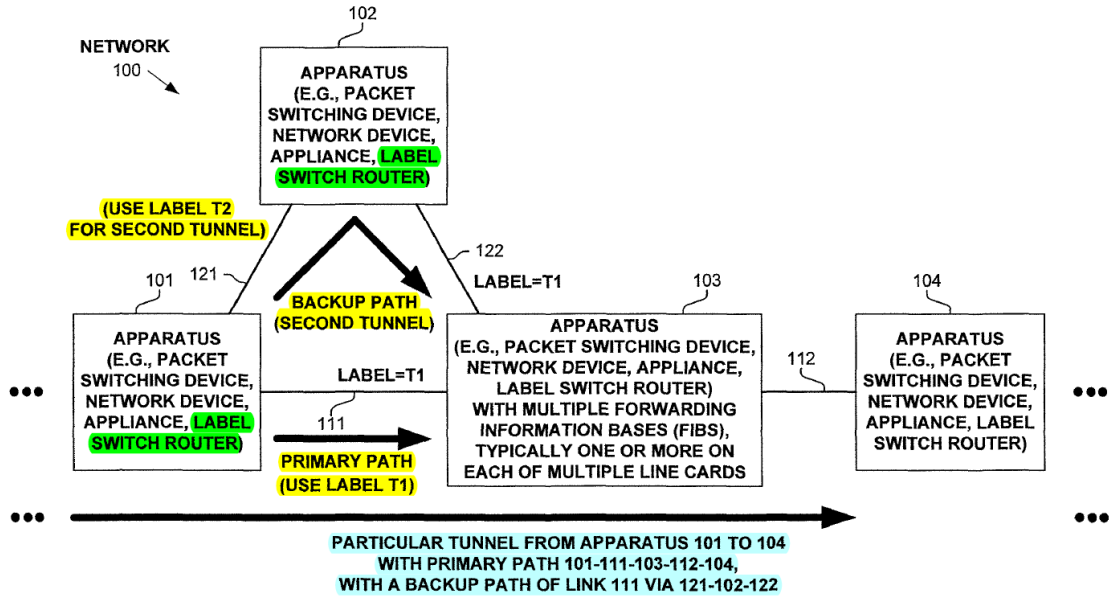
Motivations to combine include at least the similarity in subject matter between the references to the extent they concern methods of selecting paths in a network and/or determining the cost of potential paths in a network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

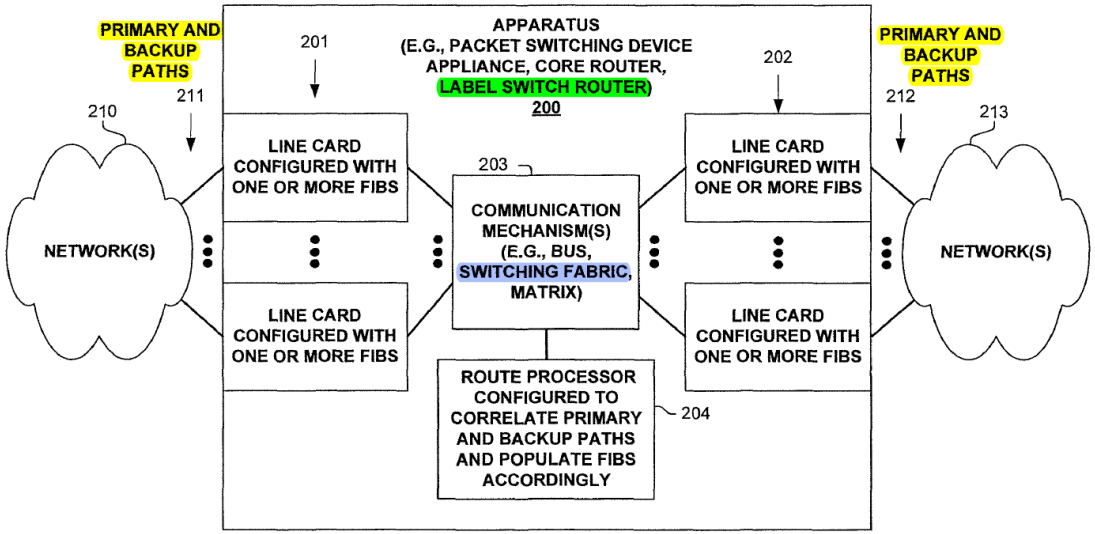
These invalidity contentions are based on Defendant's present understanding of the Asserted Claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| 1[preamble] | An entity selection method performed by a network device, comprising the steps of: | The Reference discloses an entity selection method performed by a network device.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |
| 1[a] | providing a plurality of multiprotocol label switching (MPLS) transport entities between a first endpoint and a second endpoint; | The Reference discloses providing a plurality of multiprotocol label switching (MPLS) transport entities between a first endpoint and a second endpoint.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Cisco created and developed the MPLS and MPLS-TE standards and patented technology based on those standards *before* Orckit utilized such technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Filsfils<br>• Taylor<br>• Vasseur '879[1]<br>• Rustogi |

[1] Vasseur '879 is the parent to the CON patent application that became Vasseur '460. Since Vasseur '879 and share substantially identical specifications, the citations to Vasseur '879 for each of the claim limitations herein also apply to Vasseur '460.

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | **Filsfils discloses**:<br>"In one embodiment, forwarding information bases (FIBs) are selectively populated in a packet switch. A packet switching device determines, based on one or more protocol signaling messages, a subset, which is less than all, on which FIBs a lookup operation may be performed for identifying forwarding information for a received particular packet. The packet switching device populates each of these FIBs, but not all of the FIBs of the packet switching device, with forwarding information corresponding to the particular forwarding value. Thus, FIB resources are consumed for only those FIBs which could actually be used, and not all of the FIBs, for forwarding packets in the data plane of the packet switching device, whether these packets are received on a primary or backup path." Filsfils, Abstract.<br>  <br><br>**FIGURE 1**<br>Filsfils, FIG. 1 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIGURE 2<br>Filsfils, FIG. 2 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|

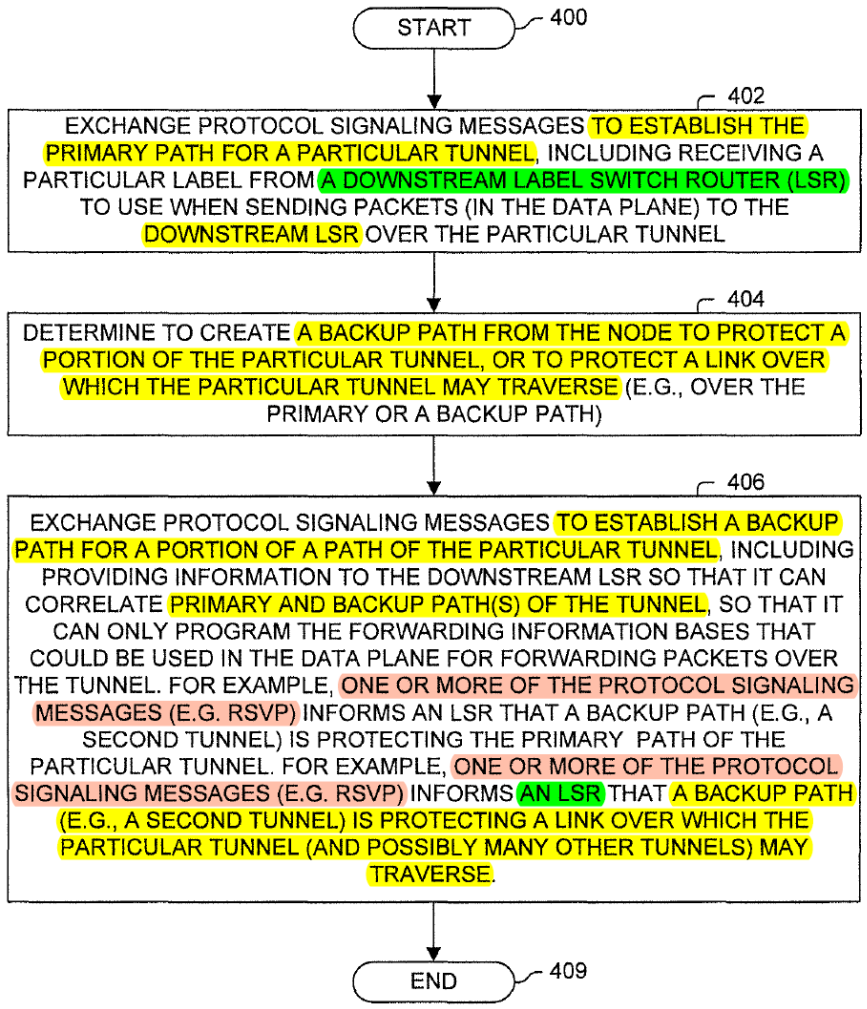
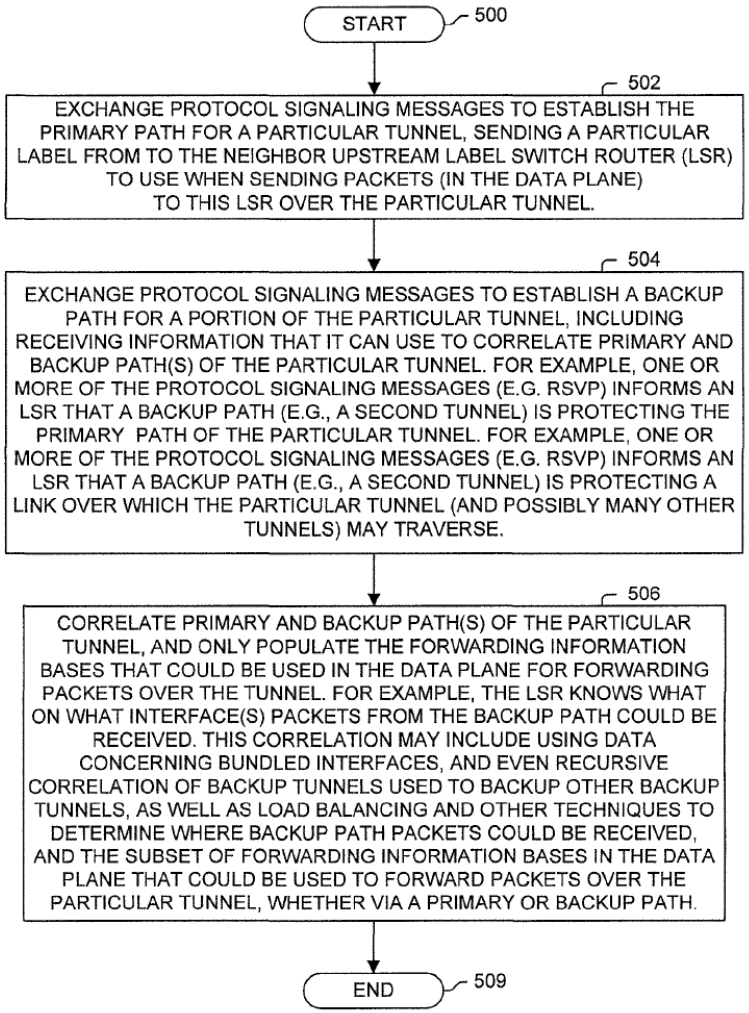



**FIGURE 4**

Filsfils, FIG. 4 (annotated).

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 5**<br>Filsfils, FIG. 5. |

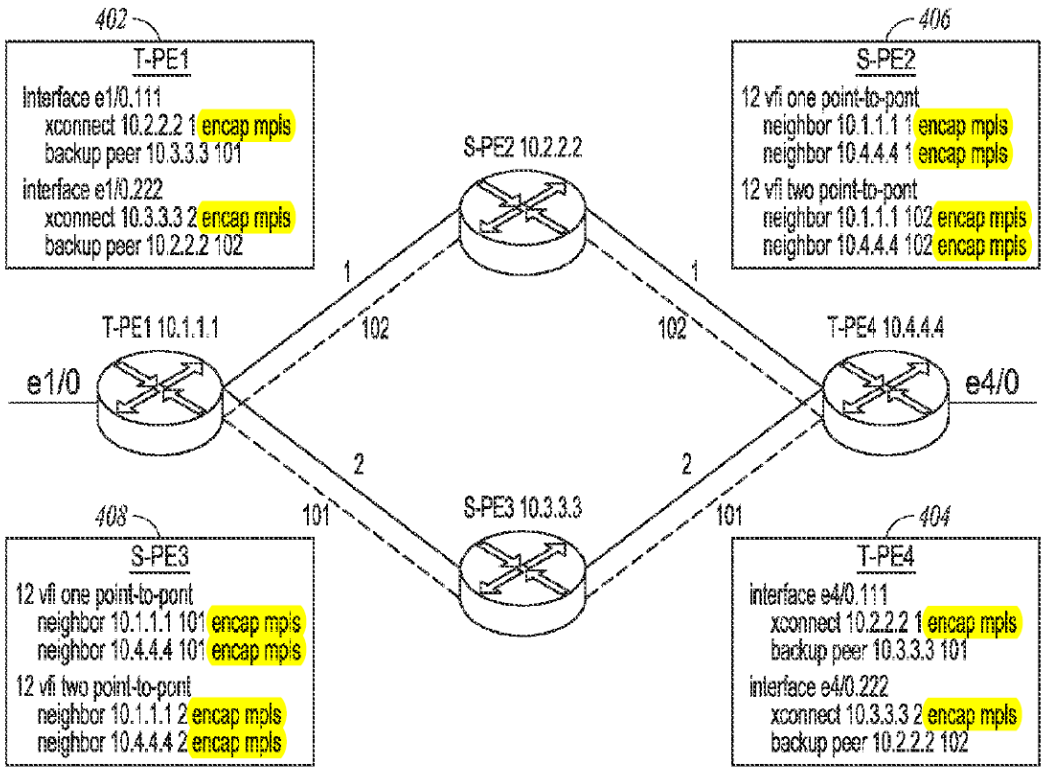
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology, including switching packets on labels especially in the core network using Multiprotocol Label Switching (MPLS)." Filsfils, 1:12-22.<br><br>"Tunnels, such as MPLS-TE (Traffic Engineering) and MPLS-TP (Transport Profile), are paths established through a network in order to transport packets efficiently through a label switched network. Fast Re-Route (FRR) is a technology that allows backup paths to be established in the network, which can be used in case of a problem with a primary path (original primary path or currently used backup path) of the tunnel. RFC 4090, entitled "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," provides an extension of the protocol signaling to establish backup label switched path (LSP) tunnels for local repair of LSP tunnels." Filsfils, 1:23-33.<br><br>"Expressly turning to the figures, FIG. 1 illustrates a network 100 operating according to one embodiment. Shown are four apparatus 101-104 (e.g., packet switching devices such as a label switch router, network device, and/or appliance). For explanation purposes, each of apparatus 101-104 will be referenced as a label switch router (LSR)." Filsfils, 5:41-46.<br><br>"As shown, a particular tunnel is established, using a signaling protocol and exchanging of protocol signaling messages. Note, LSR 101 may, or may not, be an endpoint of the particular tunnel (e.g., LSR 101 may be an intermediate LSR on the path of the particular tunnel). The primary path of the particular tunnel includes spans from LSR 101 via link 111 to LSR 103 and via link 112 to LSR 104. Note, LSR 104 may be an intermediate LSR on the path of the particular tunnel, or an endpoint of the particular tunnel. Further, for this example embodiment, LSR 103 signals LSR 101 to use label T1 at the top of the label stack in the header of a packet sent to it on the particular tunnel." Filsfils, 5:47-58.<br>"A second tunnel from LSR 101 via link 121 to LSR 102 and via link 122 to LSR 103 is similarly configured using a signaling protocol and exchanging of protocol signaling |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | messages. For example purposes, LSR 102 signals LSR 101 to use label (T2) at the top of the label stack in the header of a packet sent to it on the second tunnel. In one embodiment, LSR 101 creates the second tunnel in response to determining, or being instructed to, create a backup path to protect link 111 and/or protect all or certain tunnels traversing link 111." Filsfils, 5:59-67.<br><br>"As shown in FIG. 1, link 111 (primary path of the particular tunnel and/or all or certain tunnels traversing link 111) is protected by LSR 101 using the second tunnel (backup path). When sending packets over the particular tunnel over link 111, LSR 101 includes label T1 at the top of the label stack of these packets. If link 111 cannot be used for communicating packets of the particular tunnel, LSR 101 sends packets over the backup path for the particular tunnel by sending packets to LSR 102, with these packets having a label stack including: label T2 followed by label T1. Thus, LSR 102 will receive these packets, pop the top label (T2) from the label stack of each of these packets, identify based on label T2 to send these packets to LSR 103. After popping the top label, the label at the top of the label stack of these packets is T1, which is the same label LSR 103 expects to receive for the particular tunnel. Therefore, these packets received with label T1 at the top of their label stack, are forwarded (after popping label T1 from their label stack) by LSR 103 over the particular tunnel to LSR 104." Filsfils, 6:6-24.<br><br>"One embodiment acquires such additional information by extending Resource Reservation Protocol (RSVP) to provide information which allows a packet switch to correlate primary and backup paths. Thus, a packet switch can use this additional information in determining which of its forwarding information bases (FIBs) could possibly be used in forwarding packets (e.g., in the data plane of the packet switch)." Filsfils, 6:51-57.<br><br>"In providing this additional information to LSR 103, one embodiment communicates an extended RSVP message (including a new or modified RSVP object) or other message to LSR 103 on the second tunnel. This messages designates one or more primary tunnels (e.g., label T1 in our example) and/or a link (e.g., link 111). As LSR 103 knows what interface that it received this message, LSR 103 knows that it must populate forwarding information for these primary tunnels, either specified (e.g., by a label such as T1), or all labels corresponding to |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | tunnels which could be received over link 111. In one embodiment, the extended RSVP or other message communicated to LSR 103 also includes an identification of the backup tunnel (e.g., T2) over which the RSVP or other message is being received, as the identification the tunnel over which a packet is received is often not communicated in a packet (e.g., in the case of Penultimate Hop Popping)." Filsfils, 7:19-34.<br><br>"As shown in FIG. 1, one embodiment includes apparatus 103, which populates less than all of its FIBs with forwarding information for a tunnel (although all FIBs may be populated for certain tunnels). One embodiment includes apparatus 101 and/or 102 which communicates, via a signaling protocol (e.g., an extension of RSVP, or using another protocol), information which allows apparatus 103 to determine the relationship between primary and backup paths, such that apparatus 104 can correlate this primary and backup path information (possibly also correlating backup path of backup path information, and/or bundled interfaces and/or bundled links) to identify a minimum subset of the FIBs that could possibly be used in forwarding packets of particular primary paths (e.g., tunnels)." Filsfils, 7:49-62<br><br>"Turning to FIG. 2, illustrates an apparatus 200 (e.g., packet switching devices such as a label switch router, network device, and/or appliance) operating in one embodiment. As shown, apparatus 200 includes line cards 201, 202 communicatively coupled via communication mechanism(s) 203 (e.g., bus, switching fabric, and/or matrix). Additionally, route processor 204 is configured to correlate primary and backup paths of tunnels, and to populate minimum subsets of FIBs with forwarding information for labels. Again, a minimum subset of FIBs for a particular path or label of the particular path is the set of FIBs that are determined to possibly be used in forwarding packets of a primary path, whether the label is received in a packet over the primary path or over a backup path, and possibly considering backup paths of a backup path and/or the possibly effect of bundled interfaces and/or bundled links." Filsfils, 7:63-8:11.<br><br>"As shown in FIG. 2, apparatus 200 is communicatively coupled via primary and backup paths 211, 212 to networks 210 and 213 (which could be the same network). As illustrated, each of line cards 201, 202 includes one or more FIBs. By correlating on which line card(s) 201, 202 and even within line cards 201, 202 that have multiple FIBs, primary and backup path(s) of tunnels, the number of FIB entries populated in apparatus 200 can typically be reduced, |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | possibly significantly saving memory/storage resources and resources used to populate the FIBs." Filsfils, 8:12-21.<br><br>"FIG. 4 illustrates a process performed in one embodiment. Processing begins with process block 400. In process block 402, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes receiving a particular label for a downstream LSR to use when sending packets to the downstream LSR over the particular tunnel." Filsfils, 8:61-67.<br><br>"In process block 404, a determination is made to create a backup path from the node (e.g. the node performing these operations). This backup path may be used to protect one or more particular tunnels, and/or may be used to protect a link which is used to carry packet traffic of one or more tunnels." Filsfils, 9:1-5.<br><br>"In process block 406, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including providing information to the downstream LSR so that the downstream LSR can correlate primary and backup path(s) of the particular tunnel and substantially only program the FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:6-23.<br><br>"FIG. 5 illustrates a process performed in one embodiment. Processing begins with process block 500. In process block 502, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes sending a particular label for an upstream LSR to use when sending packets over the particular tunnel to this apparatus (e.g., an LSR performing these operations)." Filsfils, 9:26-32. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "In process block 504, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including receiving information that the LSR can use to correlate primary and backup path(s) of the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:33-47.<br><br>"In process block 506, the primary and backup path(s) of the particular tunnel are correlated to identify the set of FIBs that could possibly be used in forwarding packets of the particular tunnel. Substantially only those FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel (either through a primary or backup path) are populated with the forwarding information (e.g., an entry corresponding to the label it advertised to use for the particular tunnel) for the particular tunnel. For example, the LSR knows what on what interface(s) packets from the backup path could be received. This correlation may include using data concerning bundled interfaces, and even recursive correlation of backup tunnels used to backup other backup tunnels, as well as load balancing and other techniques to determine where backup path packets could be received, and the subset of forwarding information bases in the data plane that could be used to forward packets over the tunnel, whether via a primary or backup path." Filsfils, 9:48-65.<br><br>**Taylor discloses**:<br>"Grouping pseudowires based on hardware interfaces and configured control paths enables improved pseudowire failover performance. Signaling status changes (e.g., from standby to active status) is facilitated by using group IDs for the pseudowire groups, thereby enabling improved failover performance when there is disruption in the network." Taylor, Abstract. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Taylor, FIG. 4 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|

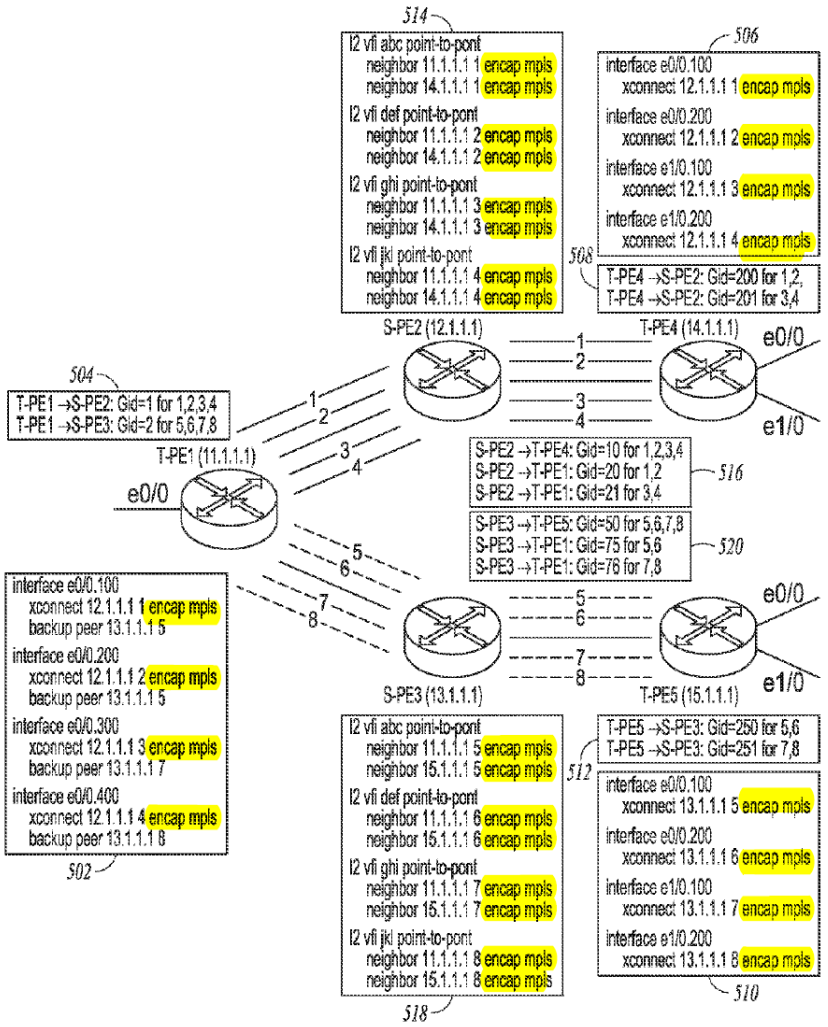
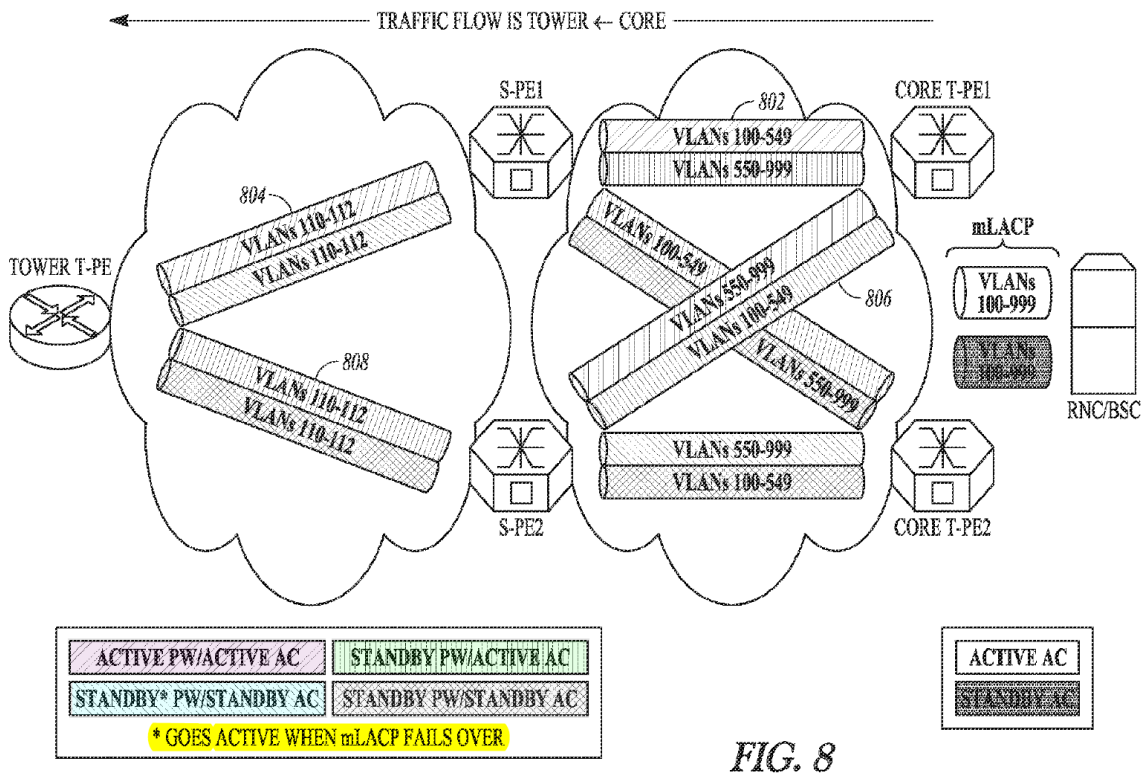
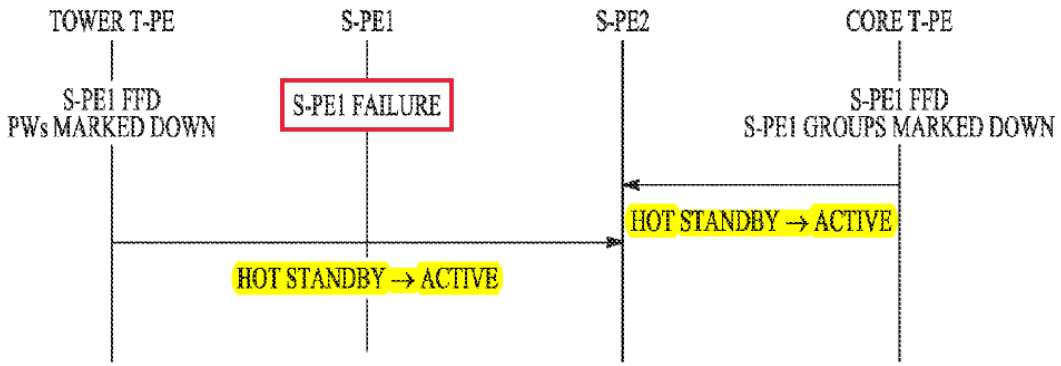
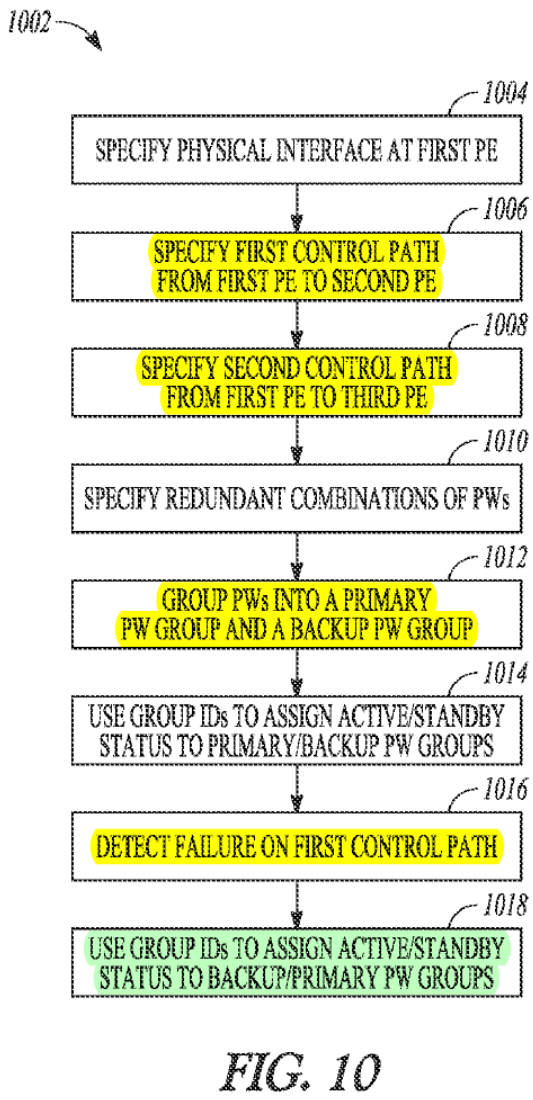



*FIG. 5*

Taylor, FIG. 5 (annotated).

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br>Taylor, FIG. 8 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Taylor, FIG. 9 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Taylor, FIG. 10 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "The present disclosure relates generally to communication networks and more particularly to pseudowire configurations in communication networks." Taylor, 1:8-10.<br><br>"Virtual Private Network (VPN) services provide secure network connections between different locations. A company, for example, can use a VPN to provide secure connections between geographically dispersed sites that need to access the corporate network so that each customer edge (CE) end point or node can communicate directly and independently with all other CE nodes. Different types of VPNs have been classified by the network layer used to establish the connection between the customer and provider network. For example, Virtual Private LAN Service (VPLS) is an architecture that delivers a multipoint Layer 2 VPN (L2VPN) service that in all respects emulates an Ethernet Local Area Network (LAN) across a wide metropolitan geographic area. All services in a VPLS appear to be on the same LAN, regardless of location. In other words, with VPLS, customers can communicate as if they were connected via a private Ethernet segment, i.e., multipoint Ethernet LAN services." Taylor, 1:12-28.<br><br>"In this context, each CE device at a customer site is connected to the service provider network at a provider edge PE) device by an Attachment Circuit (AC) that provides the customer connection to a service provider network, that is, the connection between a CE node and its associated PE node. Within the provider network, each PE device includes a Virtual Switch Instance (VSI) that emulates an Ethernet bridge (i.e., switch) function in terms of Media Access Control (MAC) address learning and forwarding in order to facilitate the provisioning of a multipoint L2VPN. A pseudowire (PW) is a virtual connection between two PE devices that connect two attachment circuits. In the context of the VPLS service, a pseudowire can be thought of as a point-to-point virtual link for each offered service between a pair of VSIs. Therefore, if each VSI can be thought of as a virtual Ethernet switch for a given customer service instance, then each pseudowire can be thought of as a virtual link connecting these virtual switches to each other over a Packet Switched Network (PSN) for that service instance." Taylor, 1:29-47. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Since the failure of pseudowires obviously degrades network performance, some effort has been directed towards adding system redundancies including redundant pseudowires. However, the presence of redundant pseudowires alone is insufficient to improve overall failover performance, that is, the ability to switch over automatically to a redundant or backup system. Thus, there is a need for improved methods for managing pseudowires to facilitate pseudowire switching and enable improved failover performance." Taylor, 1:48-56.<br><br>"FIG. 4 shows details for PW connectivity in an exemplary network for an example embodiment." Taylor, 1:66-67.<br><br>"FIG. 5 shows details for PW grouping for an example embodiment." Taylor, 2:1-2.<br><br>"FIG. 8 shows an example network including redundant PW connections for an example embodiment." Taylor, 2:9-10.<br><br>"FIG. 9 shows an example sequence diagram for a failure mode related to the embodiment shown in FIG. 8." Taylor, 2:11-12.<br><br>"FIG. 10 shows a flowchart that illustrates a method of providing improved PW grouping according to an example embodiment." Taylor, 2:13-15.<br><br>"According to one embodiment, a method of providing improved pseudowire performance includes specifying a physical interface at a first PE node in a network, a first control path from the first PE node to a second PE node in the network, and a second control path from the first PE node to a third PE node in the network. With these specifications, the method then includes specifying redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node, and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Then these pseudowires can be grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. Group identifiers for the primary PW group and the backup PW group can then be used to assign an |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths. The method may then include detecting a failure on the first control path, and in response to the detected failure, using the group identifiers to assign the active status to the backup pseudowires and the standby status to the primary pseudowires." Taylor, 2:27-52.<br><br>"Pseudowires are used in pseudowire emulation edge-to-edge to provide a Layer 2 Virtual Private Network (L2VPN) connection. When large numbers (e.g., 4,000-6,000) of pseudowires are aggregated together on a single router, failure performance tends to be linear or O(n) where n is the number of pseudowires. While O(n) performance may be acceptable for small numbers of pseudowires, the effect on network outages can be increasingly undesirable as the number of pseudowires increases." Taylor, 2:54-62.<br><br>"For example, a cell-site router will typically start an approximately 2-minute procedure if contact with its controller, which is reached via a pseudowire, is lost for more than some threshold amount (e.g., between approximately 0.75 and 1.75 seconds in some cases). This can be a major impediment to the scalability of pseudowire deployments. These issues have become increasingly relevant as providers of Multiservice Broadband Networks (MBNs) are rapidly replacing or augmenting their traditional Synchronous Optical Networking (SONET) equipment with cheaper Ethernet equipment in the evolution towards a 4G (i.e., 4$^{th}$ generation) network." Taylor, 2:63-3:7.<br><br>"One aspect of a solution to the problem of pseudowire failure is the deployment of redundant pseudowires. For example, redundant pseudowires have been used in the context of Multiprotocol Label Switching (MPLS) networks, which use a Label Distribution Protocol (LDP) to manage labels for forwarding traffic between routers. In this context, general requirements for redundancy schemes have been developed so that duplicate pseudowires are available when a given pseudowire fails (e.g., by using active/standby status indicators). In addition, more specific implementations for redundant pseudowires have also been developed." Taylor, 3:8-18. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 1 shows a reference network model 102 with applications to example embodiments disclosed herein. The reference network model 102 includes an aggregation network 104 of PE nodes and a distribution network 106 of PE nodes between a radio network controller (RNC) (or base station controller (BSC)) 108 on the core side of the model 102 and a radio tower 111 on the tower side of the model 102. Switching provider edge nodes S-PE1 and S-PE2 connect the two networks 104, 106. On the core side, two core terminating provider edges T-PE1 and T-PE2 connect to the RNC/BSC 108 through attachment circuits 110, 112. On the tower side, one tower terminating provider edge T-PE connects to the radio tower 111 through an attachment circuit 114." Taylor, 3:19-31.<br><br>"Additionally as noted in FIG. 1, peer-PE monitoring is carried out within each network 104, 106. That is, there is peer-PE monitoring between provider edges that share a segment, for example, by multi-hop bidirectional forwarding detection (BFD). Alternatively, peer monitoring can be accomplished by other means (e.g., MPLS-TP (Transport Protocol) keep-olives). This peer-PE monitoring is used to provide the mechanism for fast failure detection. Once a failure is detected, the network can react by "rerouting" the failed pseudowires to pre-provisioned backup paths and thus provide a minimal disruption in service to the end-user. This rerouting can be accomplished by LDP signaling between provider edges." Taylor, 3:32-44.<br><br>"The reference network model 102 may be considered as part of a larger hub-and-spoke model as shown in FIG. 2. A hub-and-spoke distribution model 202 includes a core network 204, distribution networks 206, and aggregation networks 208. Network elements including distribution nodes, aggregation nodes, and towers are also shown with nominal count values (e.g., 30 distribution nodes between the core network 204 and a distribution network 206). In this model 202, tower T-PEs are the spokes white core-PEs constitute the hub. Dozens to hundreds of tower T-PEs connect to a few S-PEs; these S-PEs are quite similar to ASBRs as they act as forwarders between the two distinct MPLS domains, providing isolation and, in the case of mobility, aggregation services. Typically, several aggregation networks 208 are connected to a single distribution network 206, eventually connecting the tower with the core router that connects the tower's ACs to the RNC/BSC. There are typically several distribution |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | networks in a Radio Access Network (RAN) connected to the service provider's core Internet Protocol (IP) network." Taylor, 4:11-30.<br><br>"With reference to FIG. 1, FIG. 3 shows a variety of failure modes encountered in the reference network model 102. Failure 302 of communications between tower T-PE and the S-PE can be detected via peer monitoring when both the S-PE and the T-PE are still active/alive. For example, this failure may be due to a toss of connectivity when the BFD session goes down. Failure 304 of S-PE1 can be due to a hardware failure, power outage, or the lack of BED-session maintenance capability (e.g., if the BFD-session hello timers cannot be serviced for the prescribed period of time). Failure 306 of communications between S-PE1 and core T-PE1 can be detected via peer monitoring when both S-PE1 and core T-PE1 are still active/alive. Failure 308 at core T-PE1 can be due to a hardware failure, power outage, or the lack of BFD-session maintenance capability." Taylor, 4:46-60.<br><br>"Pseudowire connectivity is further illustrated in FIG. 4 where the illustrated network includes four nodes: T-PE1 (10.1.1.1), S-PE2 (10.2.2.2), S-PE3 (10.3.3.3), and T-PEA (10.4.4.4). For the terminating nodes T-PE1 and T-PE4, specifications for VLANs (virtual Local Area Networks) connections (i.e., pseudowires) are shown using the Internet Operating System Command Line Interface (IOS CLI). The specification 402 for T-PE1 defines two VLANS as primary/backup combinations of virtual circuits for the network. The first three lines of the specification 402 define "VLAN 111" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 111" in the first line. The second line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE2 (10.12.2) with a virtual circuit Identification (VCID) set as VCID=1, and the third line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=101. The next three lines of the specification 402 define "VLAN 222" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 222" in the fourth line. The fifth line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=2, and the sixth line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE2 (10.2.2.2) with VCID=102. These virtual circuits, VCID=1, VCID=2, VCID=101 and VCID=102 are shown in the figure between T-PE1 and the S-PEs with a solid line for the primary circuits VCID=1 |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | and VCID=2 and a dashed line for the backup circuits VCID=101 and VCID=102." Taylor, 4:61-5:23.<br><br>"With respect to T-PE1 in FIG. 4, although "VLAN 111" and "VLAN 222" share the same hardware port, they do not share the same "control path disposition." That is, "VLAN 111" is primary to S-PE2 (VCID=1) and standby to S-PE3 (VCID=101), white "VLAN 222" has an opposite configuration since it is primary to S-PE3 (VCID=2) and standby to S-PE2 (VCID=102). As discussed below, certain embodiments group pseudowires according to "control path disposition" (e.g., xconnect configuration as well as the hardware interface in order to improve failover performance. That is, to deal with both hardware port failures and switching path failures, the grouping criteria also considers the cross connects. In this case, on T-PE1 as well as T-PE4, there would exist two groups: one for active to S-PE2 and standby to S-PE3 and another for active to S-PE3 and standby to S-PE2 (i.e., the inverse configuration)." Taylor, 5:61-6:9.<br><br>"First, local connectivity is characterized by local group identifications (Group-IDs), which depend on whether the allocation is done at a T-PE or S-PE. FIG. 5 shows an embodiment that illustrates an allocation of local group IDs in a network including terminating nodes T-PE1 (11.1.1.1), T-PE4 (14.1.1.1), and T-PE5 (15.1.1.1) and switching nodes S-PE2 (12.1.1.1) and S-PE3 (13.1.1.1). The specification 502 for T-PE1 determines corresponding local group IDs 504 based on the hardware interface and the control path. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, and the Group-ID=2 corresponds to VCID=5, VCID=6, VCID=7, and VCID=8." Taylor, 6:10-21.<br><br>"Local group IDs are maintained in a database so that pseudowire redundancy is also maintained. First, in a case without pseudowire redundancy, all the xconnect configurations from the same physical interface to the same peer are assigned the same local group ID. So, for example, in Ethernet cases all xconnect configurations under sub-interfaces of the same physical interface to the same peer will be assigned the same local group ID (e.g., e0/0 and e0/1 are sub-interfaces of e0). FIG. 6 shows a database representation for T-PE4 from FIG. 5. From the root node 602 for T-PE4, there is a first interface node 604 for e0/1 and a second interface node 606 for e1/0. The first interface node 604 is configured towards a single peer |

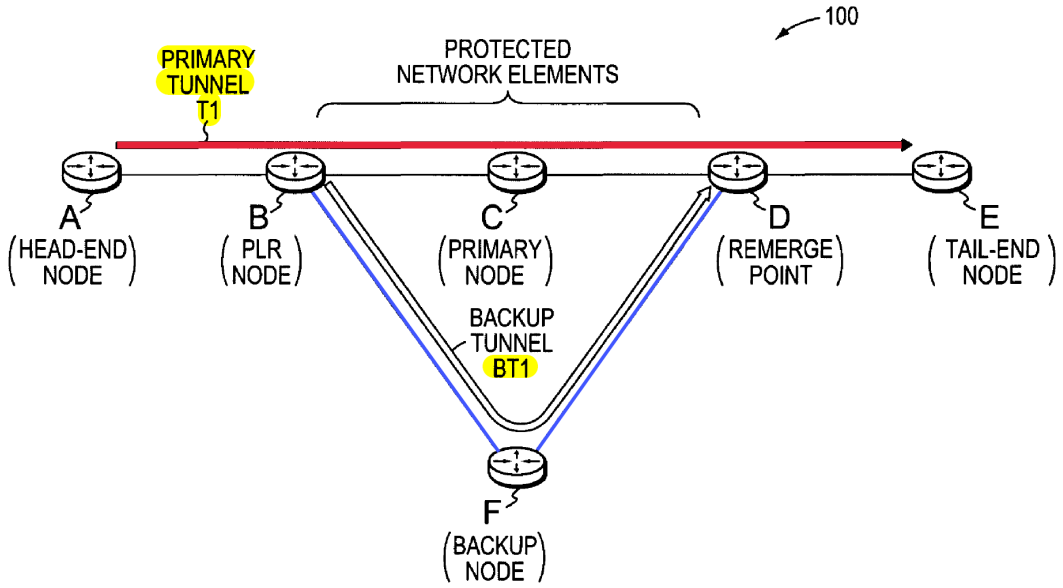
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | node (12.1.1.1) 608 and is thus assigned a single local group ID (Group-ID=200) 610. Similarly, the second interface node 606 is configured towards a single peer node (12.1.1.1) 612 and is thus assigned a single local group ID (Group-ID=201) 614. In this case from the assignment of local group IDs 508 in FIG. 5, Group-ID=200 corresponds to VCID=1 and VCID=2, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). Both pseudowires (VCID=1 and VCID=2) are assigned the same local group ID (Group-ID=200) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly, from the assignment of local group IDs 508 in FIG. 5, T-PE4 has VCID=3 and VCID=4 under the physical interface e1/0 going to the same peer S-PE2 (12.1.1.1), and the local Group ID (Group-ID=201) is assigned to these VCs." Taylor, 6:48-7:8.<br><br>"For the pseudowire redundancy case, a separate redundancy-group database is maintained by the xconnect application. This redundancy-group database contains the peer IDs in the group and the local group IDs advertised to them. This is needed to maintain a 1:1 mapping between the primary pseudowires and their corresponding backup pseudowires. FIG. 7 shows a database representation for T-PE1 from FIG. 5. From the root node 702 for T-PE1, there is an interface node 704 for e0/0 and a redundancy group node 706 that shows connections for configurations to a first peer node (12.1.1.1) 708, which is assigned a local group ID (Group-ID=1) 710, and a second peer node (13.1.1.1) 712, which is assigned a local group ID (Group-ID=2) 714. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). These pseudowires are assigned the same group ID (Group-ID=1) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly from the assignment of local group IDs 504 in FIG. 5, T-PE1 has VCID=5, VCID=6, VCID=7, and VCID=8 under the physical interface e0/0 going to another peer S-PE3 (13.1.1.1), and the local group ID (Group-ID=2) is assigned to these VCs. In this case these local group IDs are organized as a redundancy group 706." Taylor, 7:9-35.<br><br>"Pseudowire grouping allows multiple pseudowires to be aggregated together when signaling either label withdrawals or status changes between segment end-point provider edges. This signaling can be carried out through LDP grouping TLV (Type Length Value). For example, |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | when a PE node fails (e.g., failure 304 of S-PE1), aggregating the inter-segment PE signaling using the grouping TLV can provide significant scaling advantages. This allows all pseudowires sharing a physical port and PW configuration (e.g., xconnect configuration) to be signaled en masse between segment-adjacent provider edges." Taylor, 7:36-46.<br><br>"FIG. 8 shows an example based on FIG. 1 where VLAN ACs are shown as grouped by both port/HW-interface and pseudowire-class. The grouping criterion allows all "similar" pseudowires to be signaled together: All the grouped pseudowires share the same port and next-hop provider edge. Additionally, the figure contains many pseudowires, each grouped into a shaded tube. For example, the tube labeled "VLANs 100-549" contains 450 pseudowires grouped together. This figure depicts an incoming Ethernet comprised of 900 VLANs being segmented in two with 450 VLANs (100-549) active to S-PE1 while the other half of the VLANs (550-999) being active to S-PE2. This might be considered a type of manual load balancing. Furthermore, the aggregation network is only showing a single tower and the VLANs associated with it; other VLAN destinations are not shown in the figure." Taylor, 7:47-62.<br><br>"'VLANs 110-112' are active along a first pseudowire path 802 from Core T-PE1 to S-PE1 and a second pseudowire path 804 from S-PE1 to Tower T-PE. When a failure occurs at S-PE1 (e.g., as the switching node failure 304 shown in FIG. 3), then the standby pseudowires become active for 'VLANs 110-112' along a first pseudowire path 806 from Core T-PE1 to S-PE2 and a second pseudowire path 808 from S-PE2 to Tower T-PE." Taylor, 7:63-8:3.<br><br>"The standby pseudowires in FIG. 8 can be configured as HSPWs, a configuration that enables ACs to quickly failover to pre-provisioned pseudowires that are in active state but set to not-forwarding. Then when a failure occurs, switching over to these pre-provisioned HSPWs occurs quickly by switching from not-forwarding status to forwarding status." Taylor, 8:4-9.<br><br>"FIG. 9 shows a UML (Unified Modeling Language) sequence diagram of signaling events related to the failover procedure illustrated in FIG. 8 for a switching node failure 304. The Tower T-PE and the Core T-PE independently detect a failure at S-PE1 (e.g., BFD forwarding with LDP signaling), and then mark down the status of the currently active pseudowires routed |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | through S-PE1 and mark up the status of the standby pseudowires routed through S-PW2. Other failure modes shown in FIG. 3 can be handled similarly." Taylor, 8:10-18.<br><br>"In general, it is desirable for MPLS-based. Ethernet networks to react quickly to failures, so proactive detection mechanisms are employed in order to pick up system failures quickly. All proactive monitoring is typically done between PE peers on a single MPLS network. These provider edges on the edges of the MPLS network act similarly to an Autonomous System Boundary Router (ASBR). As a result, related embodiments detect control path failures, which may not be the same as pseudowire data path failures. That is, the data packets and control packets may take different paths between provider edges in a MPLS network although typically these paths are coincident. Thus, when the control and data paths are not coincident, if the control path fails, then all pseudowires utilizing the control path are marked as failed. As a corollary, if the data path fails and the control path remains healthy, then failure will not be detected from monitoring the control path." Taylor, 8:19-35.<br><br>"A failure of a monitored provider edge initiates a switchover of all active pseudowires using the failing provider edge to their configured HSPWs (if they exist). Grouping can greatly reduce the number of messages needed between provider edges (Inter-PE Aggregation) and within a single provider edge (Intra-PE Aggregation). Furthermore, the MPLS network itself may be internally resilient deploying technologies such as, but not limited to, MPLS-TE (MPLS Traffic Engineering) and ERR (Fast Reroute). The paths across the MPLS network may recover quickly and might not trip the fault-monitoring systems." Taylor, 8:36-46.<br><br>"With reference to the above discussion, FIG. 10 shows a method 1002 of providing improved PW grouping according to an example embodiment. In a first operation 1004 of the method 1002, a physical interface is specified at a first PE node in a network. In a second operation 1006, a first control path is specified from the first PE node to a second PE node in the network. In a third operation 1008, a second control path is specified from the first PE node to a third PE node in the network. These control paths related to a common physical interface can be used to characterize redundant pairs of pseudowires." Taylor, 8:48-58. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "In a fourth operation 1010, redundant combinations of pseudowires are specified, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Although a redundant combination may relate a single backup pseudowire to a given primary pseudowire, in some cases multiple backup pseudowires will be related to a given primary pseudowire for increased redundancy. In a fifth operation 1012, these pseudowires are grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. In a sixth operation 1014, group identifiers for the primary PW group and the backup PW group are used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 8:59-9:11.<br><br>"In an optional seventh operation 1016, a failure may be detected on the first control path, and in an optional eighth operation 1018, in response to the detected failure, the group identifiers may be used to assign the active status to the backup pseudowires and the standby status to the primary pseudowires. For example, the failure on the first control path may be detected by using BED packet streams between PE nodes of the network. Then the detected failure can be signaled to PE nodes in the network by sending LDP status updates between PE nodes in the network. Then, after receiving the failure detection signals, the group identifiers can be used again to assign the active status to the backup pseudowires and the standby status to the primary pseudowires by sending LDP status updates between PE nodes in the network." Taylor, 9:12-26.<br><br>"Typically the network in is an MPLS network and the PE nodes are routers that provide network services to connected CE nodes of a customer network. In general, each control path is an Internet Protocol (IP) routing path between PE nodes in the network and each data path is a label switched path (LSP) between PE nodes in the network." Taylor, 9:27-32. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 11 shows a schematic representation of an apparatus 1102, in accordance with an example embodiment. For example, the apparatus 1102 may be used to implement the method 1002 of providing improved pseudowire grouping as described above with reference to FIG. 10. The apparatus 1102 is shown to include a processing system 1104 that may be implemented on a server, client, or other processing device that includes an operating system 1106 for executing software instructions." Taylor, 10:2-10.<br><br>"In accordance with an example embodiment, the apparatus 1102 includes a PW management module 1108 that includes a first specification module 1110, a second specification module 1112, third specification module 1114, a fourth specification module 1116, a grouping module 1118, and an assignment module 1120. The first specification module 1110 operates to specify a physical interface at a first PE node in a network. The second specification module 1112 operates to specify a first control path from the first PE node to a second PE node in the network. The third specification module 1114 operates to specify a second control path from the first PE node to a third PE node in the network. The fourth specification module 1116 operates to specify redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node." Taylor, 10:11-29.<br><br>"The grouping module 1118 operates to group the pseudowires into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. The assignment module 1120 operates to use group identifiers for the PW groups to assign an active status to the primary pseudowires and a standby status to the backup pseudowires. The active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 10:30-38. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | **Vasseur '879 discloses**:<br>"A technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, Abstract.<br>  <br><br>FIG. 1<br>Vasseur '879, FIG. 1 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|

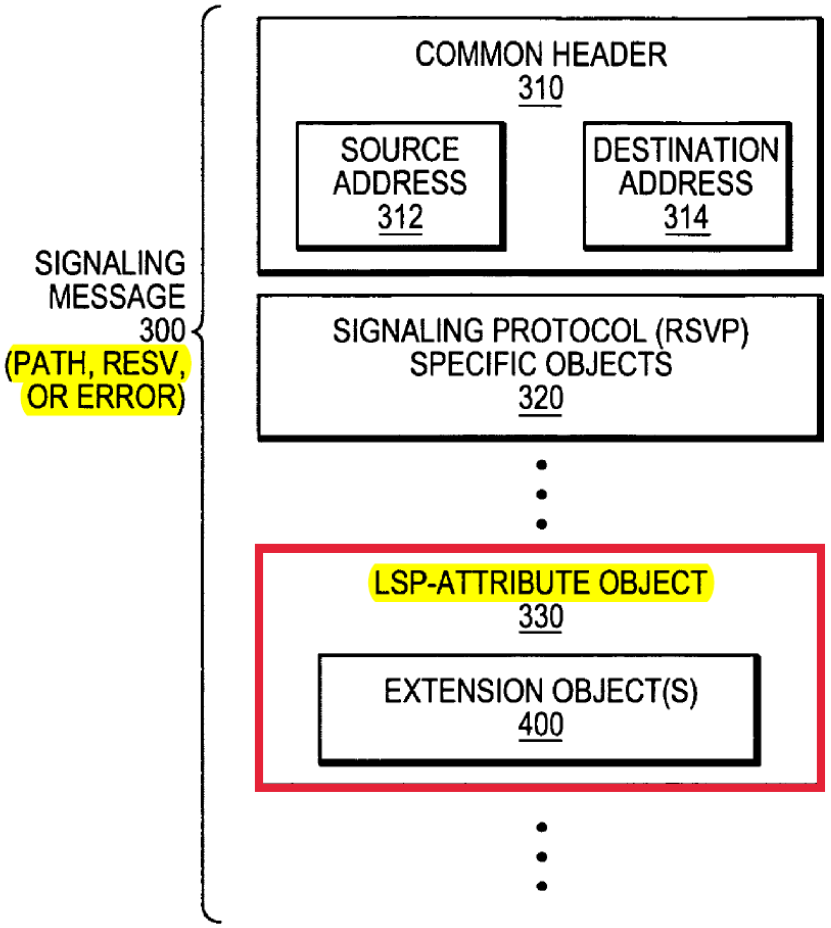
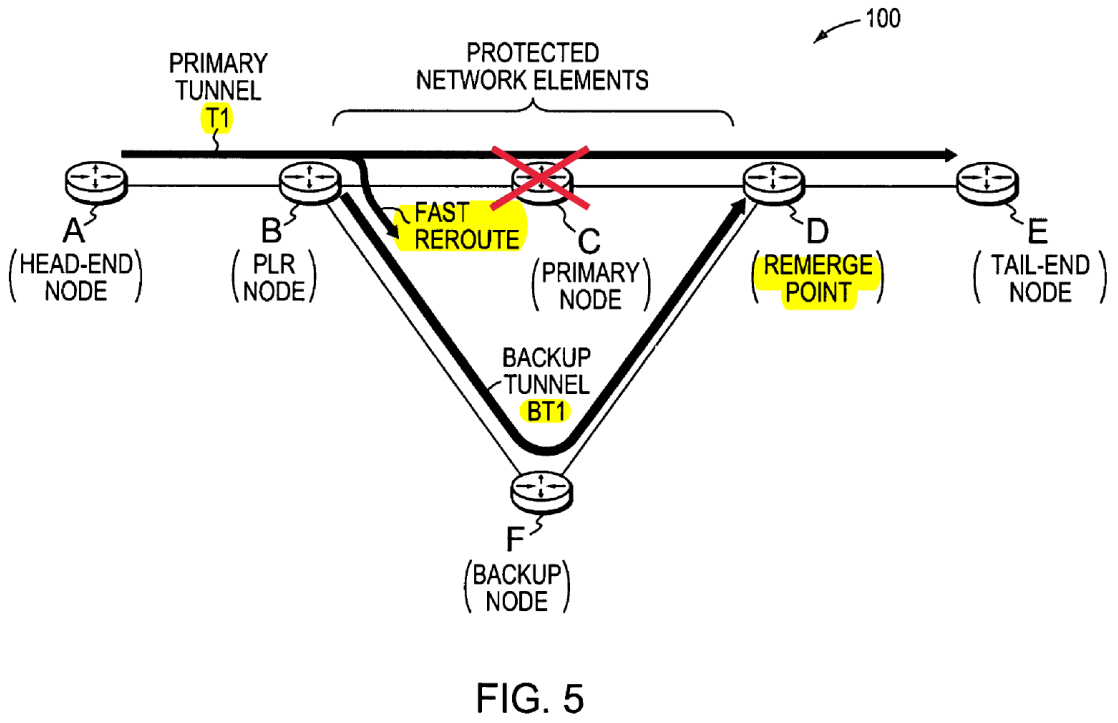
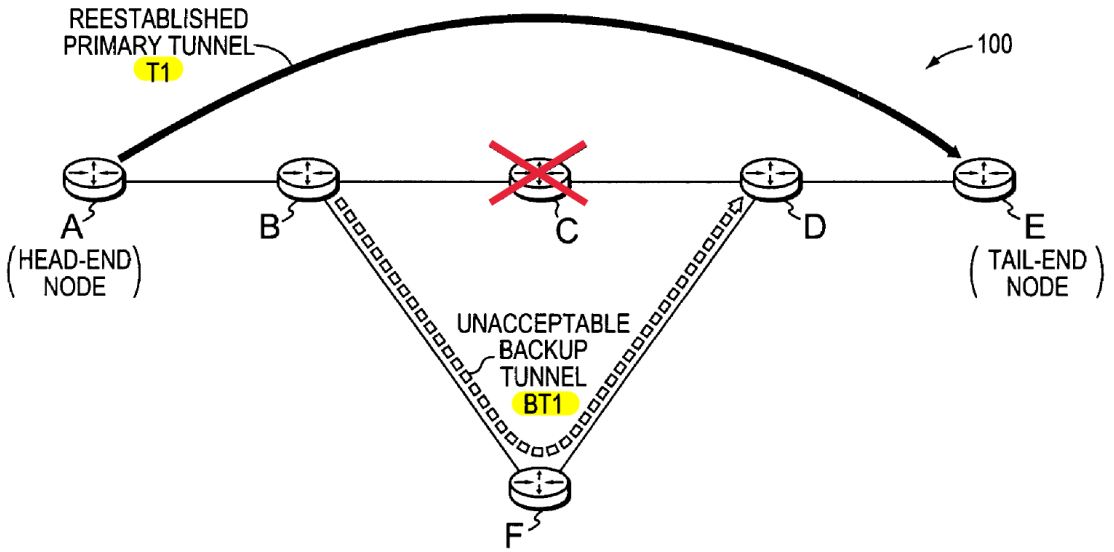
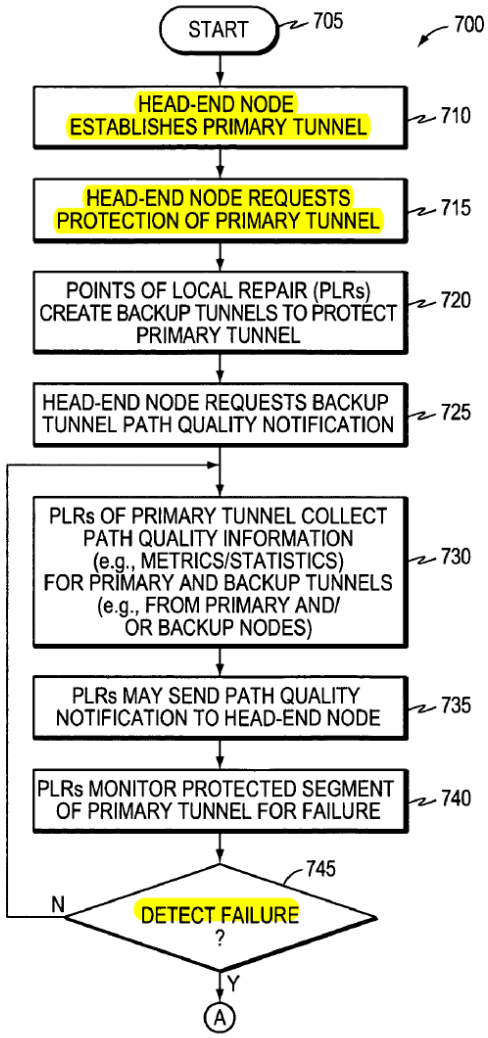


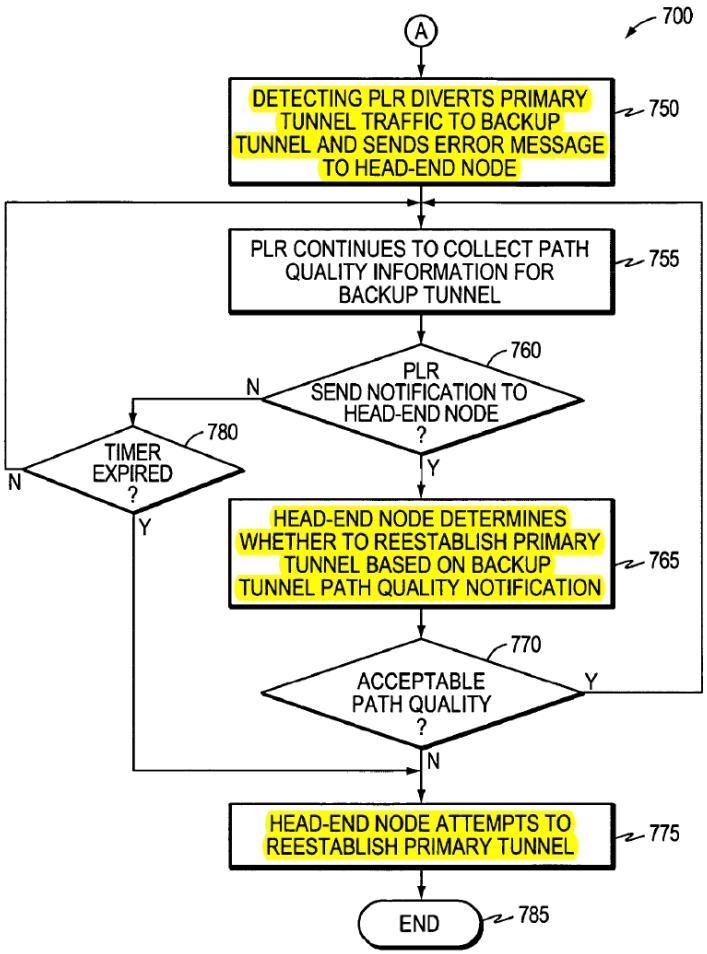

FIG. 3

Vasseur '879, FIG. 3 (annotated).

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Vasseur '879, FIG. 5 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br>Vasseur '879, FIG. 6 (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 7A<br>Vasseur '879, FIG. 7A (annotated). |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 7B<br>Vasseur '879, FIG. 7B. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "The present invention relates to computer networks and more particularly to dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 1:10-14.<br><br>"Since management of interconnected computer networks can prove burdensome, smaller groups of computer networks may be maintained as routing domains or autonomous systems. The networks within an autonomous system (AS) are typically coupled together by conventional "intradomain" routers configured to execute intradomain routing protocols, and are generally subject to a common authority. To improve routing scalability, a service provider (e.g., an ISP) may divide an AS into multiple "areas." It may be desirable, however, to increase the number of nodes capable of exchanging data; in this case, interdomain routers executing interdomain routing protocols are used to interconnect nodes of the various ASes. Moreover, it may be desirable to interconnect various ASes that operate under different administrative domains. As used herein, an AS or an area is generally referred to as a "domain," and a router that interconnects different domains together is generally referred to as a 'border router.'" Vasseur '879, 1:40-56.<br><br>"An example of an interdomain routing protocol is the Border Gateway Protocol version 4 (BGP), which performs routing between domains (ASes) by exchanging routing and reachability information among neighboring interdomain routers of the systems. An adjacency is a relationship formed between selected neighboring (peer) routers for the purpose of exchanging routing information messages and abstracting the network topology. The routing information exchanged by BGP peer routers typically includes destination address prefixes, i.e., the portions of destination addresses used by the routing protocol to render routing ("next hop") decisions. Examples of such destination addresses include IP version 4 (IPv4) and version 6 (IPv6) addresses. BGP generally operates over a reliable transport protocol, such as TCP, to establish a TCP connection/session. The BGP protocol is well known and generally described in Request for Comments (RFC) 1771, entitled A Border Gateway Protocol 4 (BGP-4), published March 1995." Vasseur '879, 1:57-2:7. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Examples of an intradomain routing protocol, or an interior gateway protocol (IGP), are the Open Shortest Path First (OSPF) routing protocol and the Intermediate-System-to-Intermediate-System (IS-IS) routing protocol. The OSPF and IS-IS protocols are based on link-state technology and, therefore, are commonly referred to as link-state routing protocols. Link-state protocols define the manner with which routing information and network-topology information are exchanged and processed in a domain. This information is generally directed to an intradomain router's local state (e.g., the router's usable interfaces and reachable neighbors or adjacencies). The OSPF protocol is described in RFC 2328, entitled OSPF Version 2, dated April 1998 and the IS-IS protocol used in the context of IP is described in RFC 1195, entitled Use of OSI IS-IS for routing in TCP/IP and Dual Environments, dated December 1990, both of which are hereby incorporated by reference." Vasseur '879, 2:8-24.<br><br>"An intermediate network node often stores its routing information in a routing table maintained and managed by a routing information base (RIB). The routing table is a searchable data structure in which network addresses are mapped to their associated routing information. However, those skilled in the art will understand that the routing table need not be organized as a table, and alternatively may be another type of searchable data structure. Although the intermediate network node's routing table may be configured with a predetermined set of routing information, the node also may dynamically acquire ("learn") network routing information as it sends and receives data packets. When a packet is received at the intermediate network node, the packet's destination address (e.g., stored in a header of the packet) may be used to identify a routing table entry containing routing information associated with the received packet. Among other things, the packet's routing information indicates the packet's next-hop address." Vasseur '879, 2:25-41.<br><br>"Multi-Protocol Label Switching (MPLS) Traffic Engineering has been developed to meet data networking requirements such as guaranteed available bandwidth or fast restoration. MPLS Traffic Engineering exploits modem label switching techniques to build guaranteed bandwidth end-to-end tunnels through an IP/MPLS network of label switched routers (LSRs). These tunnels are a type of label switched path (LSP) and thus are generally referred to as MPLS Traffic Engineering (TE) LSPs. Examples of MPLS TE can be found in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels dated December 2001, RFC 3784 |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | entitled Intermediate-System-to-Intermediate-System (IS-IS) Extensions for Traffic Engineering (TE) dated June 2004, and RFC 3630, entitled Traffic Engineering (TE) Extensions to OSPF Version 2 dated September 2003, the contents of all of which are hereby incorporated by reference in their entirety." Vasseur '879, 2:58-3:6.<br><br>"Establishment of an MPLS TE-LSP from a head-end LSR to a tail-end LSR involves computation of a path through a network of LSRs. Optimally, the computed path is the "shortest" path, as measured in some metric, that satisfies all relevant LSP Traffic Engineering constraints such as e.g., required bandwidth, "affinities" (administrative constraints to avoid or include certain links), etc. Path computation can either be performed by the head-end LSR or by some other entity operating as a path computation element (PCE) not co-located on the head-end LSR. The head-end LSR (or a PCE) exploits its knowledge of network topology and resources available on each link to perform the path computation according to the LSP Traffic Engineering constraints. Various path computation methodologies are available including CSPF (constrained shortest path first). MPLS TE-LSPs can be configured within a single domain, e.g., area, level, or AS, or may also span multiple domains, e.g., areas, levels, or ASes." Vasseur '879, 3:7-24.<br><br>"The PCE is an entity having the capability to compute paths between any nodes of which the PCE is aware in an AS or area. PCEs are especially useful in that they are more cognizant of network traffic and path selection within their AS or area, and thus may be used for more optimal path computation. A head-end LSR may further operate as a path computation client (PCC) configured to send a path computation request to the PCE, and receive a response with the computed path, potentially taking into consideration other path computation requests from other PCCs. It is important to note that when one PCE sends a request to another PCE, it acts as a PCC." Vasseur '879, 3:25-36.<br><br>"Some applications may incorporate unidirectional data flows configured to transfer time-sensitive traffic from a source (sender) in a computer network to a destination (receiver) in the network in accordance with a certain "quality of service" (QoS). Here, network resources may be reserved for the unidirectional flow to ensure that the QoS associated with the data flow is maintained. The Resource ReSerVation Protocol (RSVP) is a network-control protocol that |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | enables applications to reserve resources in order to obtain special QoS for their data flows. RSVP works in conjunction with routing protocols to, e.g., reserve resources for a data flow in a computer network in order to establish a level of QoS required by the data flow. RSVP is defined in R. Braden, et al., Resource ReSerVation Protocol (RSVP), RFC 2205, the contents of which are hereby incorporated by reference in its entirety. In the case of traffic engineering applications, RSVP signaling (with Traffic Engineering extensions) is used to establish a TE-LSP and to convey various TE-LSP attributes to routers, such as border routers, along the TE-LSP obeying the set of required constraints whose path may have been computed by various means." Vasseur '879, 3:37-57.<br><br>"Generally, a tunnel is a logical structure that encapsulates a packet (a header and data) of one protocol inside a data field of another protocol packet with a new header. In this manner, the encapsulated data may be transmitted through networks that it would otherwise not be capable of traversing. More importantly, a tunnel creates a transparent virtual network link between two network nodes that is generally unaffected by physical network links or devices (i.e., the physical network links or devices merely forward the encapsulated packet based on the new header). While one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 3:58-4:3.<br><br>"Occasionally, a network element (e.g., a node or link) will fail, causing redirection of the traffic that originally traversed the failed network element to other network elements that bypass the failure. Generally, notice of this failure is relayed to the nodes in the network through an advertisement of the new network topology, e.g., an IGP or BGP Advertisement, and routing tables are updated to avoid the failure accordingly. Reconfiguring a network in response to a network element failure using, e.g., pure IP rerouting, can be time consuming. Many recovery techniques, however, are available to provide fast recovery and/or network configuration in the event of a network element failure, including, inter alia, "Fast Reroute", e.g., MPLS TE Fast Reroute. An example of MPLS TE Fast Reroute is described in Pan, et al., Fast Reroute Extensions to RSVP-TE for LSP Tunnels, RFC 4090, May 2005, which is hereby incorporated by reference as though fully set forth herein." Vasseur '879, 4:4-21. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Fast Reroute (or FRR) has been widely deployed to protect against network element failures, where "backup tunnels" are created to bypass one or more protected network elements (e.g., links, shared risk link groups (SRLGs), and nodes). When the network element fails, traffic is quickly diverted ("Fast Rerouted") over a backup tunnel to bypass the failed element, or more particularly, in the case of MPLS, a set of primary TE-LSPs (tunnels) is quickly diverted. Specifically, the point of local repair (PLR) node configured to reroute the traffic inserts ("pushes") a new label for the backup tunnel, and the traffic is diverted accordingly. Once the failed element is bypassed, the backup tunnel label is removed ("popped"), and the traffic is routed along the original path according to the next label (e.g., that of the original TE-LSP). Notably, the backup tunnel, in addition to bypassing the failed element along a protected primary TE-LSP, also intersects the primary TE-LSP, i.e., it begins and ends at nodes along the protected primary TE-LSP." Vasseur '879, 4:22-39.<br><br>"To offer maximum protection, e.g., guaranteed bandwidth, during Fast Reroute, backup tunnels may reserve a configurable amount of bandwidth to ensure that Fast Rerouted traffic from the primary tunnel has a reserved path to follow. For example, the bandwidth reserved for the primary tunnel may also be reserved on the backup tunnel. While this approach provides maximum protection, it also requires a non-negligible amount of network resources (e.g., capacity/bandwidth) and may increase operational complexity." Vasseur '879, 4:40-48.<br><br>"Certain techniques are available to efficiently minimize the amount of resources required by the establishment and maintenance of the backup tunnels for Fast Reroute. One such technique is to create zero-bandwidth ("0-BW") backup tunnels (i.e., tunnels that reserve no bandwidth) to protect non-0-BW primary tunnels. This "best effort" approach does not guarantee that the path followed by the backup tunnel will have enough bandwidth to support the diverted primary tunnel at the time of failure without QoS degradation, however in many situations the path quality of the backup tunnel is sufficient. For instance, if the network is not overly congested, or the backup tunnel follows a non-congested path, there may be enough available bandwidth along the backup tunnel to support the newly rerouted traffic. Also, because primary tunnels often reserve bandwidth in response to "peak" traffic utilization, the amount of traffic over the primary tunnel at the time of failure may be far less than the reserved bandwidth (e.g., at "off-peak" times). Notably, those skilled in the art will understand that in |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | the absence of the above exceptions, a 0-BW backup tunnel may have unacceptable bandwidth (e.g., affecting path quality) to support the diverted traffic." Vasseur '879, 4:49-5:2.<br><br>"Currently, head-end nodes (LSRs) may be configured to systematically reroute the primary tunnels affected by the network element failure (e.g., diverted primary tunnels), especially in the case with 0-BW backup tunnels, such as, e.g., by reestablishing a new primary tunnel that follows a path excluding the failed network element. In particular, 0-BW backup tunnels represent a best effort attempt to allow the head-end node to more gracefully reestablish the primary tunnel in response to a failure, since the backup tunnels may not be able to support the diverted traffic without QoS degradation. The systematic reestablishing may potentially result in the reestablishment of a large number of primary tunnels (e.g., up to 3000 for a single network element failure in today's networks). Notably, reestablishing diverted primary tunnels may be undesirable for the network, such as by creating traffic churn, jitter, control plane overloads, etc., as will be understood by those skilled in the art. However, as noted above, there are situations where the backup tunnel may provide acceptable bandwidth, at least, for example, for a period of time (e.g., possibly short) until the failed network element is restored. In these situations, then, it may have been unnecessary to reestablish the diverted primary tunnels. There remains a need, therefore, for a technique that dynamically determines whether to reestablish a diverted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 5:3-28.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 5:32-47. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as Resource ReSerVation Protocol (RSVP) Traffic Engineering (TE) signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or type/length/value (TLV) encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object of the RSVP messages to convey the path quality notification." Vasseur '879, 5:48-58.<br><br>"In accordance with one aspect of the present invention, the head-end node requests the establishment of the primary tunnel (e.g., a TE-Label Switched Path, TE-LSP), along with a request for Fast Reroute protection of one or more network elements (e.g., with zero-bandwidth, 0-BW backup tunnels) at a PLR node. In addition, the head-end node may include a request for backup tunnel path quality notification, such as, e.g., through the use of the novel Feedback sub-object. The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and sends an error message (e.g., an RSVP PathErr) to the head end node indicating the "Fast Rerouting" of the primary tunnel." Vasseur '879, 5:59-6:6.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, the PLR node may collect metrics/statistics (e.g., packet drops, path cost, jitter, etc.) of the primary and/or backup tunnels. Once the primary tunnel is Fast Rerouted, the PLR node continues to collect metrics of the backup tunnel, and may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. Notably, the PLR node may be configured to send path quality notifications to the head-end node once, continually, periodically, in response to configurable changes in path quality, etc. Also, as in the case where multiple primary tunnels are Fast Rerouted, the path quality notification may include an indication of which Fast Rerouted primary tunnels in particular have been effected by the changed path quality." Vasseur '879, 6:7-23. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "In accordance with yet another aspect of the present invention, upon receiving the error message (PathErr), the head-end node may wait for the path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. Notably, in the event the head-end node does not receive a path quality notification for the backup tunnel (e.g., within a configurable time limit), the head-end node may attempt to reestablish the new primary tunnel. Moreover, where the head-end node has multiple primary tunnels being Fast Rerouted, a configurable subset of the primary tunnels may be reestablished, e.g., to reduce congestion of the backup tunnels, and/or to limit the number of reestablished primary tunnels within a given period of time." Vasseur '879, 6:24-43.<br><br>"Advantageously, the novel technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. By providing the head-end node of the primary tunnel with path quality feedback of the backup tunnel, the novel technique avoids reestablishing a potentially large number of tunnels over one or more alternate paths after a failure (and Fast Reroute) if the backup tunnels have acceptable path quality. In particular, the backup tunnels, e.g., 0-BW backup tunnels, may not be congested or subsequently burdened by the Fast Rerouted traffic of the primary tunnel. Also, the failed network element (thus the primary tunnel) may be quickly restored; therefore by not reestablishing the primary tunnel, network jitter, churn, etc., may be avoided. Further, the dynamic nature of the novel technique alleviates the need for cumbersome manual configuration." Vasseur '879, 6:44-59.<br><br>"FIG. 3 is schematic block diagram of an exemplary signaling (RSVP) message that may be advantageously used with the present invention." Vasseur '879, 7:6-8.<br><br>"FIG. 5 is a schematic block diagram of the computer network in FIG. 1 showing Fast Reroute protection of a primary tunnel using a backup tunnel in accordance with the present invention." Vasseur '879, 7:12-15. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 6 is a schematic block diagram of the computer network in FIG. 5 showing an unacceptable backup tunnel path quality and resultant reestablishing of the primary tunnel in accordance with the present invention." Vasseur '879, 7:16-19.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention." Vasseur '879, 7:20-24.<br><br>"FIG. 1 is a schematic block diagram of an exemplary computer network 100 comprising a plurality of nodes A-F, such as routers or other network devices, interconnected as shown. The nodes may be a part of one or more autonomous systems, routing domains, or other networks or subnetworks. For instance, routers A and E may be provider edge (PE) devices of a provider network, (e.g., a service provider network) that are interconnected to one or more customer networks through customer edge (CE) devices (not shown, while the remaining nodes B-D and F may be core provider (P) devices, as will be understood by those skilled in the art. Those skilled in the art will also understand that the nodes A-F may be any nodes within any arrangement of computer networks, and that the view shown herein is merely an example. For example, the nodes may be configured as connections to/from one or more virtual private networks (VPNs), as will be understood by those skilled in the art. These examples are merely representative. Those skilled in the art will understand that any number of routers, nodes, links, etc. may be used in the computer network 100 and connected in a variety of ways, and that the view shown herein is for simplicity." Vasseur '879, 7:29-49.<br><br>"Data packets may be exchanged among the computer network 100 using predefined network communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Asynchronous Transfer Mode (ATM) protocol, Frame Relay protocol, Internet Packet Exchange (IPX) protocol, etc. Routing information may be distributed among the routers of the computer network using predetermined Interior Gateway Protocols (IGPs), such as conventional distance-vector protocols or, illustratively, link-state protocols, through the use of IGP Advertisements." Vasseur '879, 7:50-60. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 2 is a schematic block diagram of an exemplary router 200 that may be advantageously used with the present invention, e.g., as an edge router or a core router. The router comprises a plurality of network interfaces 210, a processor 220, and a memory 240 interconnected by a system bus 250. The network interfaces 210 contain the mechanical, electrical and signaling circuitry for communicating data over physical links coupled to the network 100. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols, including, inter alia, TCP/IP, UDP, ATM, synchronous optical networks (SONET), wireless protocols, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), etc." Vasseur '879, 7:61-8:6.<br><br>"The memory 240 comprises a plurality of storage locations that are addressable by the processor 220 and the network interfaces 210 for storing software programs and data structures associated with the present invention. The processor 220 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures. A router operating system 242 (e.g., the Internetworking Operating System, or IOS™, of Cisco Systems, Inc.), portions of which is typically resident in memory 240 and executed by the processor, functionally organizes the router by, inter alia, invoking network operations in support of software processes and/or services executing on the router. These software processes and/or services may comprise routing services 247, Traffic Engineering (TE) services 244, and RSVP services 249. It will be apparent to those skilled in the art that other processor and memory means, including various computer-readable media, may be used to store and execute program instructions pertaining to the inventive technique described herein." Vasseur '879, 8:7-26.<br><br>"Routing services 247 contain computer executable instructions executed by processor 220 to perform functions provided by one or more routing protocols, such as IGP (e.g., OSPF and IS-IS), IP, BGP, etc. These functions may be configured to manage a forwarding information database (not shown) containing, e.g., data used to make forwarding decisions. Routing services 247 may also perform functions related to virtual routing protocols, such as maintaining VRF instances (not shown) as will be understood by those skilled in the art." Vasseur '879, 8:27-36. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "RSVP services 249 contain computer executable instructions for implementing RSVP and processing RSVP messages in accordance with the present invention. RSVP is described in RFC 2205, entitled Resource ReSerVation Protocol (RSVP), and in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels, both as incorporated above." Vasseur '879, 8:37-42.<br><br>"TE services 244 contain computer executable instructions for operating TE functions in accordance with the present invention. Examples of Traffic Engineering are described in RFC 3209, RFC 3784, and RFC 3630 as incorporated above, and in RFC 3473, entitled, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions dated January 2003, which is hereby incorporated by reference in its entirety. A TE database (TED, not shown) may be illustratively resident in memory 240 and used to store TE information provided by the routing protocols, such as IGP, BGP, and/or RSVP (with TE extensions, e.g., as described herein), including, inter alia, path quality information as described herein. The TED may be illustratively maintained and managed by TE services 244." Vasseur '879, 8:43-57.<br><br>"Changes in the network topology may be communicated among routers 200 using a link-state protocol, such as the conventional OSPF and IS-IS protocols. Suppose, for example, that a communication link fails or a cost value associated with a network node changes. Once the change in the network's state is detected by one of the routers, that router may flood an IGP Advertisement communicating the change to the other routers in the network. In this manner, each of the routers eventually "converges" to an identical view of the network topology." Vasseur '879, 8:58-67.<br><br>"In one embodiment, the routers described herein are IP routers that implement Multi-Protocol Label Switching (MPLS) and operate as label switched routers (LSRs). In one simple MPLS scenario, at an ingress to a network, a label is assigned to each incoming packet based on its forwarding equivalence class before forwarding the packet to a next-hop router. At each router, a forwarding selection and a new substitute label are determined by using the label found in the incoming packet as a reference to a label forwarding table that includes this information. At the network egress, a forwarding decision is made based on the incoming label |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | but optionally no label is included when the packet is sent on to the next hop. In some network configurations, one hop prior to the network egress, a penultimate hop popping (PHP) operation may be performed. Particularly, because the hop prior to the network egress (the penultimate hop) is attached to the network egress, the label is no longer needed to assure that the traffic follows a particular path to the network egress. As such, the PHP-enabled device "pops" the labels from the traffic before forwarding the traffic to the network egress, e.g., using conventional or native (IP) routing, thereby alleviating the task of removing the labels at the network egress." Vasseur '879, 9:1-23.<br><br>"The paths taken by packets that traverse the network in this manner are referred to as label switched paths (LSPs) or Traffic Engineering (TE)-LSPs. An example TE-LSP is shown as the thick line and arrow (T1) between a head-end node (router A) and a tailend node (router E) in FIG. 1. Establishment of a TE-LSP requires computation of a path, signaling along the path, and modification of forwarding tables along the path. MPLS TE establishes LSPs that have guaranteed bandwidth under certain conditions. Illustratively, the TE-LSPs may be signaled through the use of the RSVP protocol (with Traffic Engineering extensions), and in particular, RSVP TE signaling messages. Notably, when incorporating the use of PCEs (described below), the path computation request (and response) between PCC and PCE can be exchanged in accordance with a protocol specified in Vasseur, et al., Path Computation Element (PCE) Communication Protocol (PCEP)—Version 1—<draft-vasseur-pce-pcep-02.txt>, Internet Draft, September 2005, the contents of which are hereby incorporated by reference in its entirety. It should be understood that the use of RSVP or PCEP serves only as an example, and that other communication protocols may be used in accordance with the present invention." Vasseur '879, 9:24-45.<br><br>"In accordance with RSVP, to request a data flow (TE-LSP) between a sender and a receiver, the sender may send an RSVP path request (Path) message downstream to the receiver along a path (e.g., a unicast route) to identify the sender and indicate e.g., bandwidth needed to accommodate the data flow, along with other attributes of the TE-LSP. The Path message may contain various information about the data flow including, e.g., traffic characteristics of the data flow. Also in accordance with the RSVP, a receiver establishes the TE-LSP between the sender and receiver by responding to the sender's Path message with a reservation request |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | (Resv) message. The reservation request message travels upstream hop-by-hop along the flow from the receiver to the sender. The reservation request message contains information that is used by intermediate nodes along the flow to reserve resources for the data flow between the sender and the receiver, to confirm the attributes of the TE-LSP, and provide a TE-LSP label. If an intermediate node in the path between the sender and receiver acquires a Path message or Resv message for a new or established reservation (TE-LSP) and encounters an error (e.g., insufficient resources, failed network element, etc.), the intermediate node generates and forwards a path or reservation error (PathErr or ResvErr, hereinafter Error) message to the sender or receiver, respectively." Vasseur '879, 9:46-10:2.<br><br>"FIG. 3 is a schematic block diagram of portions of a signaling message 300 (e.g., RSVP message, such as Path, Resv or Error) that may be advantageously used with the present invention. Message 300 contains, inter alia, a common header 310 and one or more signaling protocol specific objects 320, such as an LSP-ATTRIBUTE object 330. The common header 310 may comprise a source address 312 and destination address 314, denoting the origination and requested termination of the message 300. Protocol specific objects 320 contain objects necessary for each type of message 300 (e.g., Path, Resv, Error, etc.). For instance, a Path message may have a sender template object, Tspec object, Previous-hop object, etc. The LSP-ATTRIBUTE object 330, for instance, may be used to signal attributes and/or information regarding an LSP (tunnel). To communicate this information, LSP-ATTRIBUTE object 330 (as well as specific objects 320) may include various type/length/value (TLV) encoding formats and/or flags, as will be understood by those skilled in the art. An example of an LSP-ATTRIBUTE object is further described in Farrel, et al., Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE <draft-ietf-mpls-rsvpte-attributes-05.txt>, Internet Draft, May 2005, which is hereby incorporated by reference as though fully set forth herein. A Resv message, on the other hand, may have specific objects 320 for a label object, session object, filter spec object, etc., in addition to the LSP-ATTRIBUTE object 330. Error messages 300 (e.g., PathErr or ResvErr), may also have specific objects 320, such as for defining the type of error, etc." Vasseur '879, 10:3-31. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "It should be noted that in accordance with RSVP signaling, the state of the TE-LSP is refreshed on a timed interval, e.g., every thirty seconds, in which RSVP Path and Resv messages are exchanged. This timed interval is configurable by a system administrator. Moreover, various methods understood by those skilled in the art may be utilized to protect against route record objects (RROs) contained in signaling messages for a TE-LSP in the event security/privacy is desired. Such RRO filtering prevents a head-end node of the TE-LSP from learning of the nodes along the TE-LSP, i.e., nodes within the provider network." Vasseur '879, 10:4-42.<br><br>"Although the illustrative embodiment described herein is directed to MPLS, it should also be noted that the present invention may advantageously apply to Generalized MPLS (GMPLS), which pertains not only to packet and cell-based networks, but also to Time Division Multiplexed (TDM) and optical networks. GMPLS is well known and described in RFC 3945, entitled Generalized Multi-Protocol Label Switching (GMPLS) Architecture, dated October 2004, and RFC 3946, entitled Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control, dated October 2004, the contents of both of which are hereby incorporated by reference in their entirety." Vasseur '879, 10:43-55.<br><br>"To obviate delays associated with updating routing tables when attempting to avoid a failed network element (i.e., during convergence), some networks have employed MPLS TE Fast Reroute (FRR). MPLS Fast Reroute is a technique that may be used to quickly divert ("Fast Reroute") traffic around failed network elements in a TE-LSP. MPLS Fast Reroute is further described, for example, by Fast Reroute Extensions to RSVP-TE for LSP Tunnels, as incorporated by reference above. According to the technique, one or more network elements (e.g. links or nodes) in a network are protected by backup tunnels following an alternate path. If a failure occurs on a protected link or node, TE-LSPs (and consequently the traffic that they carry) are locally diverted onto an appropriate alternate path (e.g., a "backup tunnel") by the node immediately upstream from the failure. The backup tunnel acts as a Fast Reroute path for the primary TE-LSP and obviates delays associated with other measures, such as tearing down the primary TE-LSP after having gracefully diverted the TE-LSPs affected by the failure, should an alternate path around the failed network element exist. In the event of a |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | failure of a protected element the head-end node of the backup tunnel (or a "point of local repair," PLR node) may quickly begin diverting traffic over the backup tunnel with minimal disruption to traffic flow. Those skilled in the art will understand that MPLS Fast Reroute is one example of link or node failure protection, and that other known correction mechanisms may be used in accordance with the present invention. As mentioned above, however, the head-end node of the Fast Rerouted primary tunnel may attempt to reestablish the primary tunnel in response to learning of the protected element failure, particularly in the case where the backup tunnel is a zero-bandwidth (0-BW) tunnel. The attempt to reestablish the primary tunnel has conventionally been a systematic response to Fast Rerouting (diverting) of the primary tunnel, regardless of the path quality of the backup tunnel." Vasseur '879, 10:56-11:23.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 11:24-39.<br><br>"In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as RSVP TE signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or TLV encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object 330 of the RSVP messages 300 to convey the path quality notification." Vasseur '879, 11:40-49. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 4 is a schematic block diagram illustrating the format of an extension object (TLV) 400, such as a Fast Reroute Feedback object, that may be advantageously used with the present invention. The extension object (or sub-object) 400 is illustratively embodied as a TLV contained in an LSP-ATTRIBUTE object 330 of an RSVP message 300 and is extended to carry backup (and/or primary) tunnel path quality information. To that end, the "Feedback" object 400 is organized to include a Type field 405 containing a predetermined type value signifying the specific content of the object 400. The Length field 410 is a variable length value. The TLV encoded format may also comprise one or more non-ordered sub-TLVs 450 carried within the TLV "payload" (e.g. Value field 415), each having a Type field 455, Length field 460, and Value field 465. The fields of the TLV 400 and sub-TLV(s) 450 are used in a variety of manners, including as described herein, according to the present invention." Vasseur '879, 11:59-12:8.<br><br>"In accordance with one aspect of the present invention, the head-end node (e.g., router A) requests the establishment of the primary tunnel (e.g., T1), such as a TE-LSP. Notably, the head-end node may be a head-end node for multiple primary tunnels, as will be understood by those skilled in the art. Along with the primary tunnel establishment, the head-end node may also request Fast Reroute protection of one or more network elements (e.g., all intermediate network elements) at a PLR node (e.g., router B as shown). Note that each intermediate node along the primary tunnel may act as a PLR node, and that router B is shown merely for simplicity. Illustratively, the Fast Reroute protection may be embodied as one or more zero-bandwidth (0-BW) backup tunnels at the PLR node (e.g., BT1). Those skilled in the art will also understand that the PLR node may protect more than one primary tunnel originating at more than one corresponding head-end node (not shown)." Vasseur '879, 12:9-25.<br><br>"The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. For example, various connectivity verification protocols, such as, e.g., Bidirectional Forwarding Detection (BFD), IGP "Hello" packets, BGP KEEPALIVE messages, etc., may be used to detect a failure of a network element, as will be understood by those skilled in the art. Furthermore, other lower-layer failure detection mechanisms (e.g. optical or SONET/SDH alarms) may be used to detect a network element failure. Once failure is detected, the PLR node diverts the |

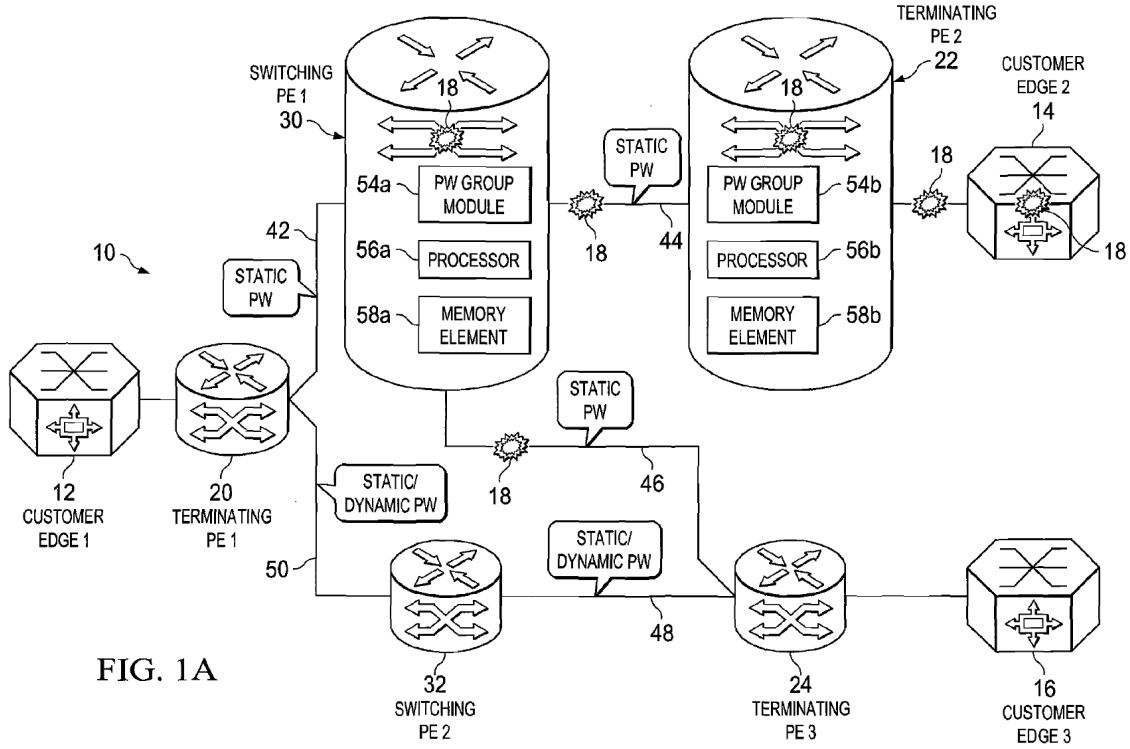
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | traffic onto the backup tunnel, and may send an error message (e.g., an RSVP PathErr 300, such as a conventional "tunnel locally repaired" message) to the head end node indicating the "Fast Rerouting" of the primary tunnel. FIG. 5 is a schematic block diagram of the computer network 100 in FIG. 1 showing Fast Reroute protection of the primary tunnel T1 (e.g., in response to a protected network element, router C, failure, indicated with an overlaid "X") using a backup tunnel BT1 in accordance with the present invention. Traffic originally received at the PLR node (router B) over the primary tunnel is now diverted over the backup tunnel to a remerge point (router D) of the primary tunnel, as will be understood by those skilled in the art." Vasseur '879, 12:42-65.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, each PLR node may collect metrics/statistics of the primary and/or backup tunnels. For instance, example metrics may comprise, inter alia, packet drops, path cost, jitter, delay, bandwidth, etc. The PLR node may collect the metrics through traffic monitoring, probes, independent calculations, and/or through cooperation with protected nodes of the primary tunnel (primary nodes) and nodes of the backup tunnel (backup nodes), e.g., transmitting path quality notifications. Once the primary tunnel is Fast Rerouted (i.e., diverted after failure of a protected network element), the PLR node continues to collect metrics of the backup tunnel. (Alternatively, metrics of the backup tunnel may be collected only after Fast Reroute, and not prior to Fast Reroute.)" Vasseur '879, 12:66-13:12.<br><br>"For example, a path cost increase from the primary tunnel to the backup tunnel may be computed (and transmitted, below) by the PLR node prior to the failure (or during the failure while the primary TE-LSP is diverted onto the backup tunnel) using its own routing tables. The path cost increase may be calculated as a difference between the entire length (head-end node to tail-end node) of the primary and backup tunnels, or just the difference between the protected segment of the primary tunnel and the backup tunnel (PLR node to remerge point). Also, a jitter increase of the primary and backup tunnels, which may be generally described as a difference between inter-arrival of packets, may be monitored using various known techniques, such as, e.g., sending probe packets (probes) from the PLR node to the remerge point. For instance, probes may determine that packets arrive at the remerge point along the primary tunnel from the PLR node consistently, e.g., every 10 milliseconds (ms) (e.g., an |

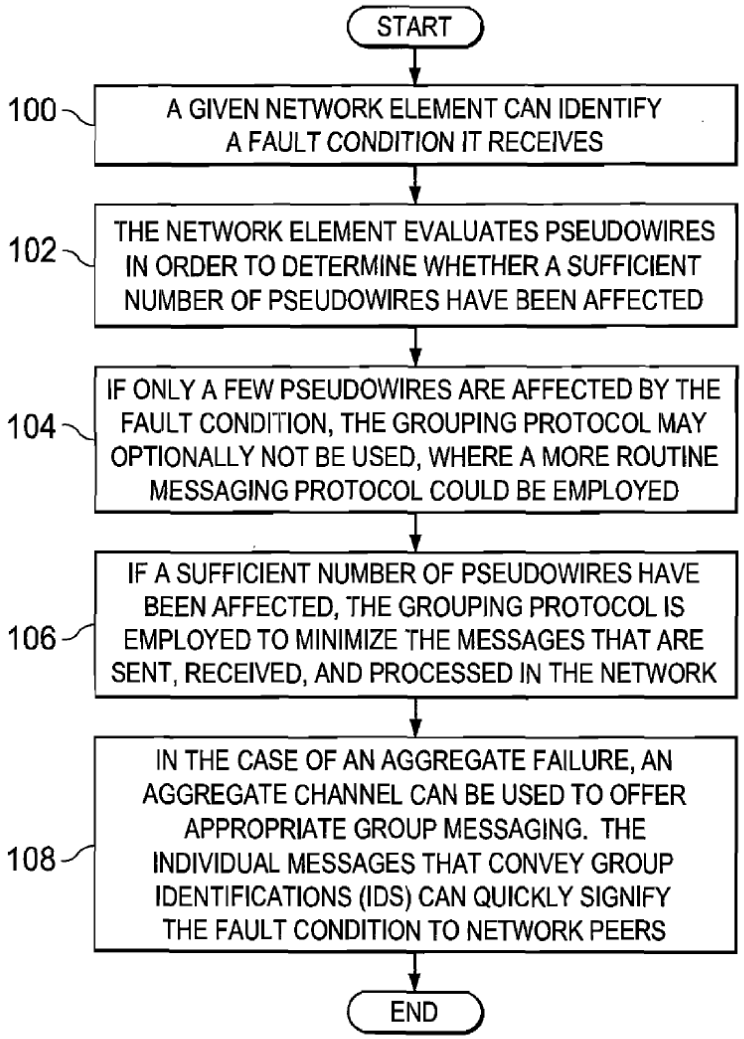
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | average value). After Fast Reroute, however, probes may determine that packets do not arrive at the remerge point along the backup tunnel from the PLR node consistently, e.g., one may arrive in 10 ms, another in 100 ms, another in 50 ms, etc. The non-constant rate of received packets (jitter) may be undesirable, e.g., in particular for voice over IP (VoIP) traffic, as will be understood by those skilled in the art." Vasseur '879, 13:13-36.<br><br>"As a further example, packet dropping may be monitored for the primary and backup tunnels prior to and after Fast Reroute (respectively). For instance, based on the tunnel label of the dropped packet, primary nodes and/or backup nodes may be able to distinguish which tunnel corresponds to the dropped packets. Each of the primary and/or backup nodes collect packet drop data, and periodically inform the PLR node of the number of dropped packets (e.g., though a corresponding Feedback object 400). In the case of a backup node, the PLR node receiving the notification may interpret the association of the backup tunnel label and the primary tunnel label to reference an appropriate primary tunnel. Those skilled in the art will understand that the above path quality metrics are merely examples, and that any other metrics/statistics useful for determining path quality of the backup tunnel may be used in accordance with the present invention (e.g., delay, bandwidth, etc.). Further, the path quality information may be measured and compared in a variety of manners, such as, e.g., as a difference between primary and backup tunnels before and after Fast Reroute, or simply the difference between the backup tunnel before and after Fast Reroute, etc." Vasseur '879, 13:37-58.<br><br>"Also after the primary tunnel is Fast Rerouted, the PLR node may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. For instance, the novel Feedback object 400 may include one or more sub-TLVs 450 corresponding to metrics/statistics, as described above. Notably, the path quality information pertaining to a particular metric/statistic may be transmitted as total values for interpretation by the head-end node (e.g., to determine the difference), or as PLR-node-computed differences (e.g., between the primary and backup tunnels, or before and after Fast Reroute). For instance, if the delay of the primary tunnel (along the protected segment) prior to Fast Reroute were 2 ms, and after Fast Reroute the delay of the backup tunnel were 5 ms, the notification may be configured to |

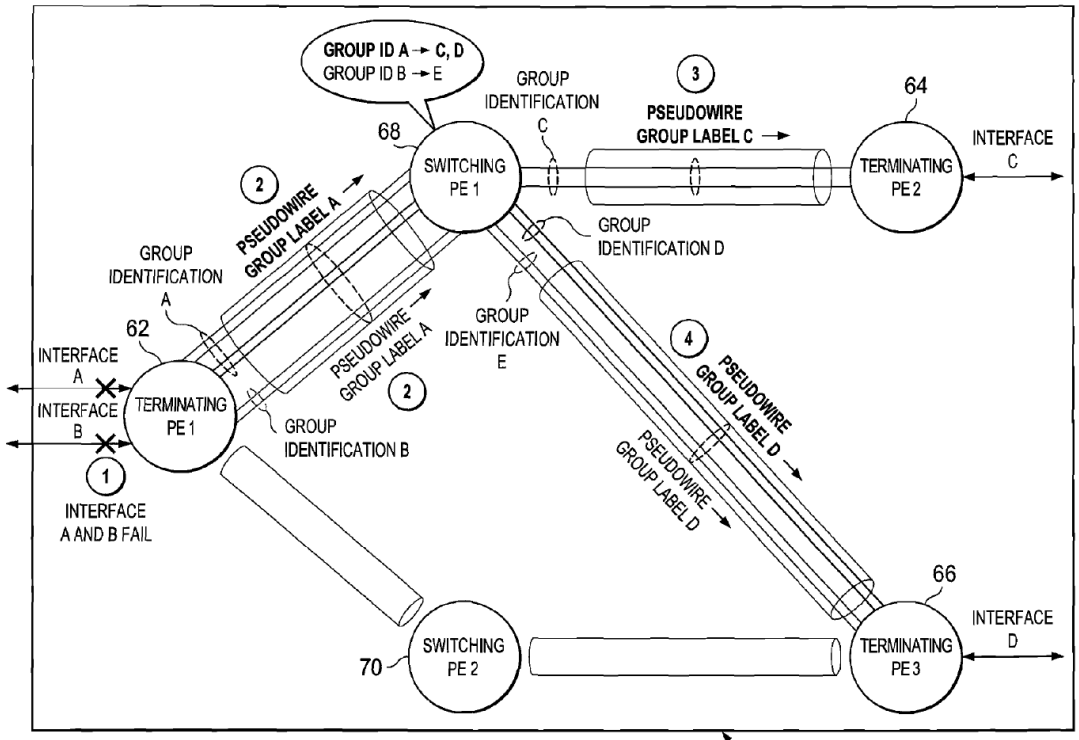
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | include both values 2 ms and 5 ms, or instead simply the difference, i.e., an increase of 3 ms." Vasseur '879, 13:59-14:8.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message 300 (PathErr), the head-end node may wait for at least one path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. Also, any number of metrics may be used in the determination, e.g., as configured by a system administrator. For example, using the metrics described above, a head-end node may be configured to reestablish the primary tunnel in response to i) a certain number of packet drops, ii) a percent increase in packet drops, iii) a number of packet drops and a percent increase in path cost, iv) a percent increase in path cost and a percent increase in jitter, etc. Those skilled in the art will understand that these are merely examples of possible path quality comparisons and reestablishment determinations, and that any comparisons to any metrics at any configurable changes may be used in accordance with the present invention." Vasseur '879, 14:41-59.<br><br>"If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. FIG. 6 is a schematic block diagram of the computer network 100 in FIG. 5 showing an unacceptable backup tunnel path quality (dotted line and arrow) and resultant reestablishment of the primary tunnel in accordance with the present invention. Those skilled in the art will understand that the reestablished primary tunnel may traverse one or more primary nodes (not shown), and that it may be computed specifically to avoid the failed network element and any network elements of the unacceptable backup tunnel. Those skilled in the art will also understand that the attempt to reestablish the primary tunnel may not be able to find an acceptable path, in which case the head-end node may continue to use the unacceptable backup tunnel or other unacceptable rerouted path." Vasseur '879, 14:60-15:9. |

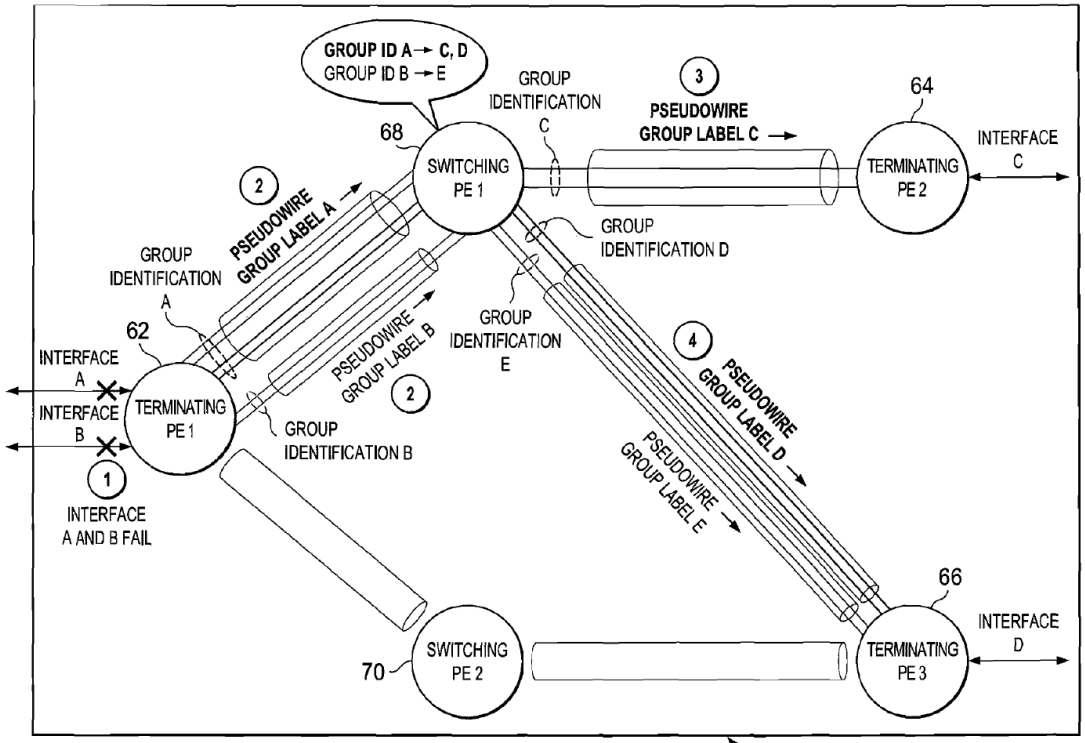
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention. The procedure 700 starts at step 705, and continues to step 710, where a head-end node (e.g., router A) establishes a primary tunnel (e.g., T1) to a destination tail-end node (e.g., router E). During or after establishment, the head-end node requests protection of the primary tunnel at step 715, and in response, PLR nodes along the primary tunnel (e.g., router B) create backup tunnels (e.g., BT1) to protect the primary tunnel in step 720. (Those skilled in the art will understand that backup tunnels around the protected network elements may already exist at the PLR node, and that "creating backup tunnels" in step 720 implies an association with pre-existing backup tunnels.) As mentioned above, these backup tunnels may illustratively be embodied as 0-BW backup tunnels. In accordance with the present invention, the head-end node may additionally request backup tunnel path quality notification from the PLR nodes in step 725, such as, e.g., through the use of empty corresponding Feedback objects in RSVP (Path) messages 300, as described above." Vasseur '879, 15:37-58.<br><br>"The procedure 700 continues to FIG. 7B (step "A"), where in step 750 the PLR node detecting the failure diverts ("Fast Reroutes") the primary tunnel traffic to the backup tunnel and sends an error message (e.g., an RSVP (Error) message 300) to the head end node, e.g., a "tunnel locally repaired" message. The detecting PLR node continues to collect path quality information for the backup tunnel in step 755 and at step 760 determines whether to send the path quality notification to the head-end node. For example, as mentioned above, the PLR node may be configured to continually send notifications, or periodically, or in response to a configurable change in path quality, etc. Also as mentioned above, the PLR node may be configured to send either the actual path quality information or the change (difference) in path quality. (As further mentioned above, metrics of the backup tunnel may alternatively be collected only after Fast Reroute, and not prior to Fast Reroute.)." Vasseur '879, 16:4-20.<br><br>"If the PLR node decides to send the notification in step 760, then the head-end node determines whether to reestablish the primary tunnel based on the backup tunnel path quality notification in step 765, e.g., based on one or more configurable thresholds, percentages, etc., as described above. If the backup tunnel is currently maintaining an acceptable quality for the |

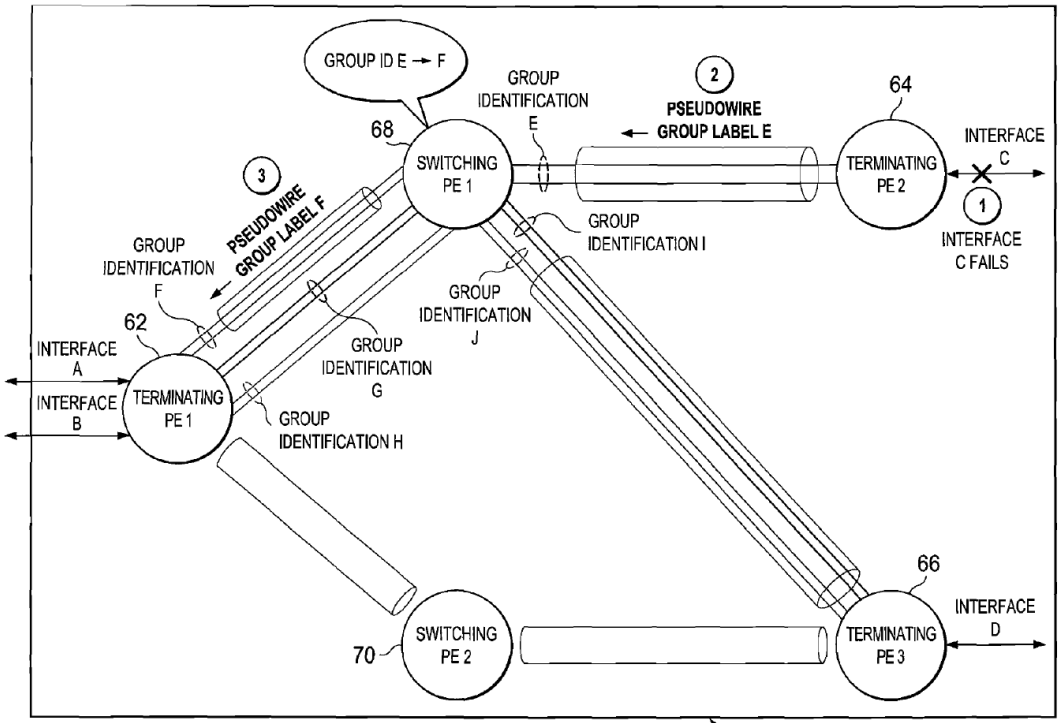
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | traffic flow in step 770, the head-end node may continue to utilize the backup tunnel, and the PLR node continues to collect path quality information in step 755 to detect any change in quality. Otherwise, if the backup tunnel quality is not acceptable in step 770, the head-end node may attempt to reestablish the primary tunnel in step 775. Notably, as mentioned above, if the PLR node has not sent any notification (step 760) within a configurable period of time in step 780, e.g., due to a backup tunnel failure, over-congestion, etc., then the head-end node may also attempt to reestablish the primary tunnel in step 775 accordingly. Moreover, as described above, in the event more than one primary tunnel is Fast Rerouted for the head-end node, various techniques to reestablish one or more of the primary tunnels may be used (e.g., as many tunnels as necessary, a configurable subset of tunnels, all tunnels, the congested tunnels, etc.). The procedure 700 ends in step 785." Vasseur '879, 16:21-43.<br><br>"While there has been shown and described an illustrative embodiment that dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the present invention. For example, the invention has been shown and described herein using "Fast Reroute," e.g., MPLS TE Fast Reroute (FRR). However, the invention in its broader sense is not so limited, and may, in fact, be used with other network element protection and failure correction mechanisms as will be understood by those skilled in the art. Moreover, while the above description describes performing the technique at the head-end node and PLR node, the invention may also be advantageously used with PCEs. In addition, while one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 16:63-17:13.<br><br>**Rustogi discloses**:<br>"An example method includes identifying a fault condition in a network, and evaluating pseudowires affected by the fault condition in order to make a determination as to whether an aggregate failure occurred in the network for a group of pseudowires. The method also includes communicating a group message indicating that the group of pseudowires is associated with the fault condition. The group message includes a group identification (ID), |

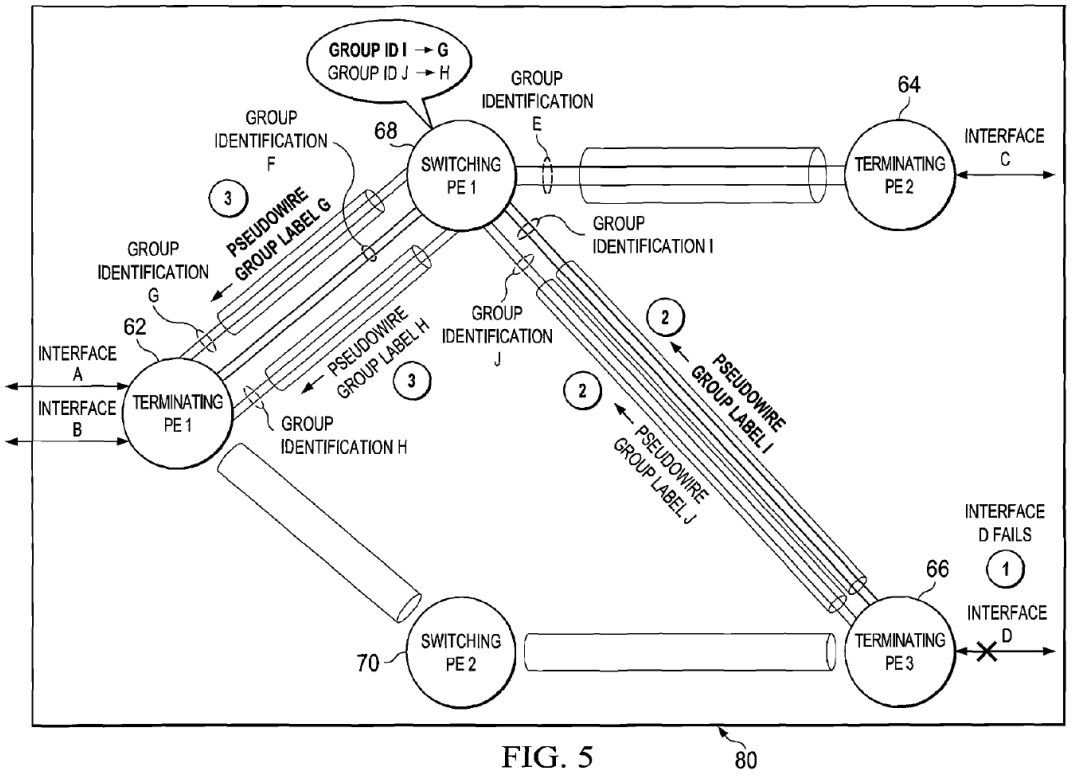
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | which identifies the group of pseudowires, and the group message includes a pseudowire group label identifying an in-band aggregate channel. More specifically, the pseudowire group label can be applicable to static pseudowires. In more detailed embodiments, the group ID identifies the group of pseudowires that are associated with an attachment circuit, a label switched path, or a port. Internal mappings can be maintained such that a plurality of pseudowires is mapped to individual interfaces of network elements in the network." Rustogi, Abstract.<br>  <br><br>FIG. 1A<br>Rustogi, FIG. 1A. |

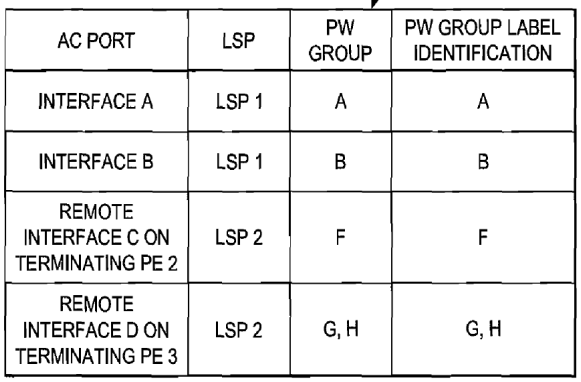
| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 1B<br>Rustogi, FIG. 1B. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Rustogi, FIG. 2. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Rustogi, FIG. 3. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Rustogi, FIG. 4. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Rustogi, FIG. 5. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | 74<br>  <br>FIG. 6<br>Rustogi, FIG. 6.<br><br>"The field of communications has become increasingly important in today's society. In particular, the ability to quickly and to effectively provision connections presents a significant challenge to component manufacturers, system designers, and network operators. Multiprotocol Label Switching (MPLS) is a mechanism in telecommunications networks that carries data from one network node to the next. Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be emulated over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDUs) and transmitting them over pseudowires. Protocols exist for establishing and maintaining the pseudowires. Certain issues have arisen in pseudowire scenarios, where faults are detected in the network." Rustogi, ¶ [0002].<br><br>"FIG. 1A is a simplified block diagram of a communication system for providing pseudowire group labels in a network environment in accordance with one embodiment of the present disclosure." Rustogi, ¶ [0004]. |

| AC PORT | LSP | PW GROUP | PW GROUP LABEL IDENTIFICATION |
|---|---|---|---|
| INTERFACE A | LSP 1 | A | A |
| INTERFACE B | LSP 1 | B | B |
| REMOTE INTERFACE C ON TERMINATING PE 2 | LSP 2 | F | F |
| REMOTE INTERFACE D ON TERMINATING PE 3 | LSP 2 | G, H | G, H |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 1B is a simplified flowchart depicting one possible, generic operational flow associated with the communication system." Rustogi, ¶ [0005].<br><br>"FIG. 2 is a simplified block diagram of an example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0006].<br><br>"FIG. 3 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0007].<br><br>"FIG. 4 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0008].<br><br>"FIG. 5 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0009].<br><br>"FIG. 6 is a simplified table of an example set of pseudowire group provisioning parameters in accordance with one embodiment." Rustogi, ¶ [0010].<br><br>"FIG. 1A is a simplified block diagram of a communication system 10 for providing pseudowire group labels in accordance with one example of the present disclosure. FIG. 1A includes a customer edge 1 (CE1) 12, a CE2 14, and a CE3 16, where a number of faults 18 are shown as propagating in the network. Typically, when an error or a failure occurs in the network (e.g., an interface failure, a pulled cable, a switch failure, hardware/software failures generally, etc.), messages are sent to various network devices in order to inform them of these fault conditions. Faults 18 of FIG. 1A are indicative of such messages, where the underlying fault condition (being signaled by the messages) can occur virtually anywhere in a network (e.g., in a customer edge, in provider equipment, etc.). FIG. 1A also includes a terminating provider equipment 1 (TPE1) 20, a TPE2 22, a TPE3 24, a switching provider edge 1 (SPE1) 30, and a SPE2 32. In one particular example implementation, the TPEs and SPEs of FIG. 1A are switches that are configured to exchange data in a network environment." Rustogi, ¶ [0012]. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "SPE1 30 may include a pseudowire (PW) group module 54 a, a processor 56 a, and a memory element 58 a. In a similar fashion, TPE2 22 may include a pseudowire group module 54 b, a processor 56 b, and a memory element 58 b. FIG. 1A also includes a number of static pseudowires 42, 44, and 46. A set of static/dynamic pseudowires 48, 50 is also provided. Note that the group labeling protocol discussed herein can be executed between individual SPEs, TPEs, or between any combinations of these elements." Rustogi, ¶ [0013].<br><br>"In one particular arrangement, communication system 10 is provided in conjunction with a Layer-2 virtual private networks (L2VPN)/operation, administration, and maintenance (OAM) L2VPN/OAM framework. The OAM framework is intended to provide OAM layering across L2VPN services, pseudowires, and packet switched network (PSN) tunnels. Communication system 10 may include any suitable networking protocol or arrangement that provides a communicative platform for communication system 10. Thus, communication system 10 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Communication system 10 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs." Rustogi, ¶ [0014].<br><br>"Failure detection and failure notification for static pseudowires is inadequate, where sluggish signaling can result in poor scalability for failover performance. Typically, static pseudowires are manually configured at respective endpoints, where control channels are absent for providing group level signaling messages. Aggregate channels are significant tools for providing suitable scalability in the network, but no such aggregate channel exists for static pseudowires. For dynamic pseudowires, such an aggregate channel may be present in the form of a label distribution protocol (LDP) directed session. However, no such protocol exists for static pseudowire configurations such that an in-band aggregate channel would be available for static pseudowires." Rustogi, ¶ [0016]. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Communication system 10 can address the aforementioned issues (and others) by offering a pseudowire group label that can represent an aggregate channel for groups of static pseudowires. The aggregate channel of communication system 10 can allow for improved scalability of failover performance. In accordance with one potential configuration of communication system 10, a pseudowire group label is representative of a group of static pseudowires transported over a label switched path (LSP). The pseudowire group label can identify the aggregate channel, which captures the hierarchy relevant to OAM mechanisms. Additionally, the groups represented by the group identification (ID) can be mutually exclusive, where a pseudowire is part of only one group. In other embodiments, a pseudowire can be part of multiple groups, or be configured in any other suitable manner based on particular network arrangements." Rustogi, ¶ [0017].<br><br>"During operations, and with brief reference to FIG. 1B, a given network element can identify a fault condition it receives (at step 100) and, subsequently, evaluate pseudowires in order to determine whether a sufficient number of pseudowires have been affected. This is reflected by step 102. If only a few pseudowires are affected by the fault condition, the grouping protocol outlined herein may have only nominal value, where there could be an option to simply communicate the fault condition in a more routine manner, as outlined in step 104. However, if a sufficient number of pseudowires have been affected, the grouping protocol outlined herein can be employed to minimize the messages that are sent, received, and processed in the network. This is reflected as step 106. Note that the determination (as to whether a sufficient number of pseudowires have been impacted by the fault condition) can involve accessing internal tables such that a quick mapping can occur to determine if an aggregate failure has occurred. As used herein, the term 'aggregate failure' simply connotes that a sufficient number of pseudowires have experienced the fault condition such that an aggregate channel can be employed to offer appropriate group messaging. For the aggregate failure condition, the individual messages that convey Group identifications (IDs) can quickly signify the fault condition to network peers, as shown in step 108." Rustogi, ¶ [0018]. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "In specific regards to OAM mechanisms, OAM messages typically result from common failures in the network. These fault conditions can be aggregated such that they are signaled as a single message, which could represent a group of failed pseudowires (as opposed to sending individual messages for each failed pseudowire). Hence, a single message could be sent to represent all the relevant OAM messages propagating in communication system 10. The group label that propagates in communication system 10 provides an architecture with a significant level of aggregation for failed pseudowires (i.e., pseudowires being affected by a given fault condition), particularly for OAM messaging. Moreover, the in-band aggregate channel of communication system 10 is based (at least in part) on the evolving trends of OAM mechanisms, which are required to be fast, responsive, and capable of being implemented in hardware or software. Additionally, in-band OAM protocols are a better measure of the path availability." Rustogi, ¶ [0020].<br><br>"In operation of one example implementation, a group label can represent the tuple <attachment circuit (AC) port level grouping, LSP>. This could signify that all pseudowires on an AC port (sought for aggregation) traverse a given LSP. Multiple pseudowire groups can exist within an LSP. Similarly, pseudowires on the same AC port (that traverse a different LSP) can use a different pseudowire group label. Alternatively, an administrator may seek to employ a one-to-one mapping between an LSP and a group label. If that were the case, then only one pseudowire group would exist within an LSP. In scenarios where there is no LSP label in the packet (e.g., due to penultimate hop popping), the pseudowire group label can provide the hierarchy that is appropriate." Rustogi, ¶ [0021].<br><br>"In one particular example, the group level pseudowire OAM message can be sent with the following label stack: Explicit/Implicit LSP Label+pseudowire group Label+GAL+ACH+pseudowire OAM with grouping TLV (where GAL=Generic Associated Channel Label, ACH=Associated Channel Header, TLV=Type-Length-Value). If there are multiple LSPs, then one group label can be provisioned for each LSP (for each pseudowire group), where per group messages can be sent on each LSP. The group label does not necessarily have a one-to-one mapping to the grouping of pseudowires implied by the Group ID in the grouping TLV. Note also that the group-based aggregate channel is applicable to |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | static pseudowires, as well as for dynamic pseudowires in certain applications." Rustogi, ¶ [0022].<br><br>"As discussed herein, the aggregate channel of communication system 10 can be configured in various ways. For example, and with regards to a first option, a separate label may simply be used to identify a pseudowire group within an LSP. The association of an OAM message and a pseudowire group is straightforward. There could potentially be multiple pseudowire group labels per LSP. As a second option, one group label can be used to identify a common pseudowire group channel on the LSP. In this implementation, one pseudowire group label is provided per LSP. The OAM message association to a pseudowire group is not as simple as the first option. As a third option, one pseudowire is simply designated to convey grouping information (e.g., without using a group label). In this case, there is no need for a pseudowire group label. Again, the OAM message association to a pseudowire group is not as simple as the first option." Rustogi, ¶ [0023].<br><br>"Any combination of formatting (for the Group ID and the pseudowire group label) can be used in the group message to be communicated in the network. In one example, only one of these elements is communicated when an aggregate fault condition is detected, or these elements can be combined into a single unique identifier. In the most generic example, a group message would at least include the Group ID (identifying the pseudowires affected by the fault) and a pseudowire group label (identifying an aggregate channel for communicating the group message). In this generic sense, a pipe (the Group ID) within a pipe (the pseudowire group label) is being identified, where the group message is identifying both elements during an aggregate fault condition. Operational details of communication system 10 are described below with reference to FIGS. 2-6. Note that before turning to additional example flows and example embodiments of the present disclosure, a brief overview of the infrastructure of communication system 10 is provided." Rustogi, ¶ [0024].<br><br>"CE1 12, CE2 14, and CE3 16 represent devices, infrastructure, equipment, clients, or customers seeking to initiate a data session in communication system 10. These elements may can comprise a digital subscriber line access multiplexer (DSLAM), a router, a personal computer, a server, a switch, and/or other devices associated with data propagation. Further, |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | these elements may sit behind, or in front of, one or more of these identified devices. The term 'CE' may be inclusive of the devices identified above (e.g., a DSLAM, a switch, etc.), as well as devices used to initiate a communication, such as a console, a proprietary endpoint, a telephone, a cellular telephone, a bridge, a computer, a personal digital assistant (PDA), a laptop or an electronic notebook, or any other device, component, element, or object capable of initiating voice, audio, media, or data exchanges within communication system 10. The customer element may also include any device that seeks to initiate a communication on behalf of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating a voice, a video, text, or a data exchange within communication system 10. Data, as used herein in this document, refers to any type of video, numeric, voice, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another." Rustogi, ¶ [0025].<br><br>"SPE1 30, SPE2 32, TPE1 20, TPE2 22, and TPE3 24 are network elements that facilitate communications in two directions in a network environment. In one particular example, each of these network elements is a switch configured to exchange data over static and/or dynamic pseudowire links. Further, the traffic exchanged between these components may be directed over an MPLS transport in certain embodiments. As used herein in this Specification, the term 'network element' is meant to encompass switches, routers, bridges, gateways, servers, processors, loadbalancers, firewalls, or any other suitable device, component, element, or object operable to exchange or process information in a network environment. Moreover, these network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information. Along similar design alternatives, any of the internal modules and components of these network elements may be combined in various possible configurations." Rustogi, ¶ [0029]. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "Turning to FIG. 2, FIG. 2 is a simplified block diagram of an example system 60 for providing an example use case using per-label switched path (LSP) pseudowire group labels. FIG. 2 includes a TPE1 62, a TPE2 64, a TPE3 66, a SPE1 68, and a SPE2 70. Each pseudowire group is identified, where a group identification (ID) for Group A and Group B is depicted at TPE1 62. Similarly, Groups C, D, and E have Group IDs at SPE1 68. TPE2 64 and TPE3 66 can couple to interfaces C and D, respectively." Rustogi, ¶ [0031].<br><br>"In this particular example, interfaces A and B have failed. Note that there is a multitude of attachment circuits (e.g., 1000 attachment circuits) that are being transported over these interfaces A and B, where the attachment circuits are being tunneled into a corresponding number of pseudowires. For example, there could be 500 attachment circuits on interface A (implicating 500 pseudowires) and 500 attachment circuits on interface B, where the fault condition for the pseudowires should be signaled. In other flawed systems, an architecture would individually signal this fault condition for each pseudowire (e.g., via signaling between TPE1 62 and SPE1 68). Instead of sending 500 messages, a single message can be sent, where a single label (and Group ID) can be used to identify the pseudowires. In this case, the Group ID A is used to signal the fault condition for 300 pseudowires and for 200 pseudowires (i.e., the top two links connecting TPE1 62 and SPE1 68) using a single message (that includes Group Label A and Group ID A). Thus, the status for Group A is quickly communicated to SPE1 68. Similarly, Group ID B can be used to signal the status of the other 500 pseudowires to appropriately convey the status for Group B. More specifically, the message can include Group Label A and Group ID B. Note that all 1000 pseudowires have effectively been accounted for using these Group IDs A and B." Rustogi, ¶ [0032].<br><br>"FIG. 3 is a simplified block diagram of an example system 72 for providing another use case for pseudowire group labels. Note that the grouping mechanism outlined herein is not limited to pseudowires that propagate over LSPs. Certain pseudowires can propagate over an LSP and represent one group, where two ports can be provisioned for two different groups (e.g., Group A and Group B). Hence, FIG. 3 is depicting a use case using pseudowire group labels for <port, LSP>mapping. In a general sense, such a configuration is showing how pseudowire mechanics can be used to offer different group signaling, which may be based on various possible implementations. Thus, there is a group level construct corresponding to the group |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | labels that are created such that any OAM protocol can send the appropriate aggregate messages. In this particular example, the signaling for Group ID A, B, C, and D is similar to that of FIG. 2; however, the grouping mechanism has simply changed." Rustogi, ¶ [0035].<br><br>"FIG. 4 is a simplified block diagram of an example system 76 for providing another use case for pseudowire group labels. In this particular example, interface C fails (as shown at TPE2 64). Note that the same logical flow occurs in FIG. 4 in terms of the group signaling, as previously discussed. The group labels in two directions do not have to be the same, where the groupings for the messaging are not necessarily symmetrical. In this particular example, TPE2 64 sends a status for Group E with the corresponding group label (i.e., Group ID E for 300 pseudowires), where that message will have a Group Label E and a Group ID E. Hence, this particular signaling is indicative of 300 pseudowires failing in the network. SPE1 68 can send the status for Group F (where the Group ID F is associated with 300 pseudowires) to TPE1 62, where that message includes a Group Label F and a Group ID F." Rustogi, ¶ [0036].<br><br>"FIG. 5 is a simplified block diagram of an example system 80 for providing another use case for pseudowire group labels. In this particular example, interface D fails (as shown at TPE3 66), where all 700 pseudowires fail. In one implementation, TPE3 66 does not have a 700 pseudowire Group ID. Instead, the Group IDs can correspond to 200 and 500 pseudowires, when summed together account for the 700 pseudowires. In this particular example, TPE3 66 sends one message for Group I (representing 200 pseudowires) and another message for Group J (representing 500 pseudowires) to SPE1 68. In response, SPE1 68 sends a message for Group G (representing 200 pseudowires) and another message for Group H (representing 500 pseudowires). Again, the signaling being exchanged between these elements is minimal due to the effective grouping of pseudowires. SPE1 68 also sends a single message for Group I (associated with 200 pseudowires) and Group J (associated with 500 pseudowires) to TPE3 66, which is coupled to interface D. Group ID G is associated with 200 pseudowires, whereas Group ID H is representative of 500 pseudowires." Rustogi, ¶ [0037]. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "FIG. 6 is a simplified table 74 illustrating an example set of pseudowire group provisioning parameters for TPE1 62, where these particular provisioning parameters could be relevant to the configuration of FIG. 3. At least in one generic sense, FIG. 2 can reflect one approach for mapping a PW group label to a PW Group ID, while FIGS. 3-5 can reflect a second approach for such mappings, where table 74 is associated with that second approach." Rustogi, ¶ [0038].<br><br>"In particular, table 74 illustrates the mapping between SPE1 68 and TPE1 62. The first column represents the attachment circuit port (e.g., interface A, interface B, remote interface C on TPE2 64, and remote interface D on TPE3 66). Additionally, table 74 depicts a number of LSPs, a set of pseudowire grouping labels, and a set of pseudowire Group IDs. Note that the Group IDs are provided inside the pseudowire group labels in this example such that these two columns match in table 74. Additionally, note that table 74 is merely representing some of the possible characteristics in a single direction, where different constructs could be used in the reverse direction. Note that the provisioning as discussed herein can significantly reduce messaging such that these presented concepts offer increased scalability. This is due in part to the nominal processing that occurs in the network, in contrast to the processing required to evaluate a prolific amount of signaling messages associated with particular pseudowires. Additionally, the paradigm discussed herein can afford service providers an adequate amount of downtime after a failure has occurred in the network." Rustogi, ¶ [0039]. |
| 1[b] | determining an overall cost for each entity pair of said plurality of entities: | The Reference discloses determining an overall cost for each entity pair of said plurality of entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | **<u>Kurose discloses</u>**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path.<br><br>"The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|

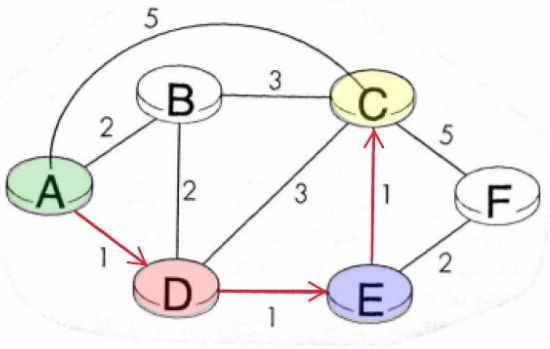



**Figure 4.4** ✦ Abstract model of a network

- the first link in the path is connected to the source
- the last link in the path is connected to the destination
- for all $i$, the $i$ and $i-1$st link in the path are connected to the same node
- for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).

In Figure 4.4, for example, the least-cost path between nodes *A* (source) and *C* (destination) is along the path *ADEC*. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)

As a simple exercise, try finding the least-cost path from nodes *A* to *F*, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from *A* to *F* by examining Figure 4.4, tracing a few routes from *A* to *F*, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between *A* and *F*? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:

Kurose at 281.

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |
| 1[c] | selecting an entity pair from said plurality of transport entities based at least in part upon said overall cost; and | The Reference discloses selecting an entity pair from said plurality of transport entities based at least in part upon said overall cost.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[b]. |
| 1[d] | if an entity pair reselection event occurs, reselecting said entity pair from the group consisting of said entity pair and a replacement entity pair comprising at least one entity distinct from the entities of said entity pair, | The Reference discloses if an entity pair reselection event occurs, reselecting said entity pair from the group consisting of said entity pair and a replacement entity pair comprising at least one entity distinct from the entities of said entity pair.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| 1[e] | wherein said entity pair reselection event is selected from a group consisting of adding an entity to said plurality of transport entities, removing an entity from said plurality of transport entities, an operational status change for one of said plurality of transport entities, and a change in overall cost for one of said plurality of transport entities. | The Reference discloses wherein said entity pair reselection event is selected from a group consisting of adding an entity to said plurality of transport entities, removing an entity from said plurality of transport entities, an operational status change for one of said plurality of transport entities, and a change in overall cost for one of said plurality of transport entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references.<br><br>**Kurose discloses**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path.<br><br>"The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|

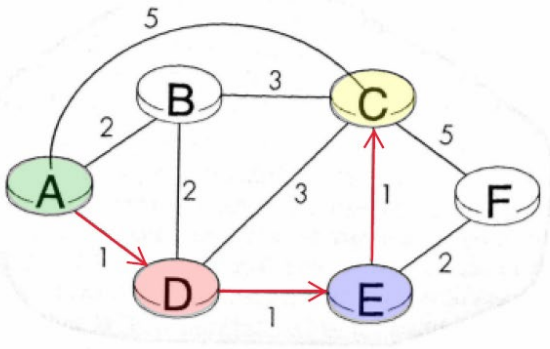



**Figure 4.4** ✦ Abstract model of a network

- the first link in the path is connected to the source
- the last link in the path is connected to the destination
- for all $i$, the $i$ and $i-1$st link in the path are connected to the same node
- for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).

In Figure 4.4, for example, the least-cost path between nodes *A* (source) and *C* (destination) is along the path *ADEC*. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)

As a simple exercise, try finding the least-cost path from nodes *A* to *F*, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from *A* to *F* by examining Figure 4.4, tracing a few routes from *A* to *F*, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between *A* and *F*? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:

Kurose at 281.

| No. | '821 Patent Claim 1 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |

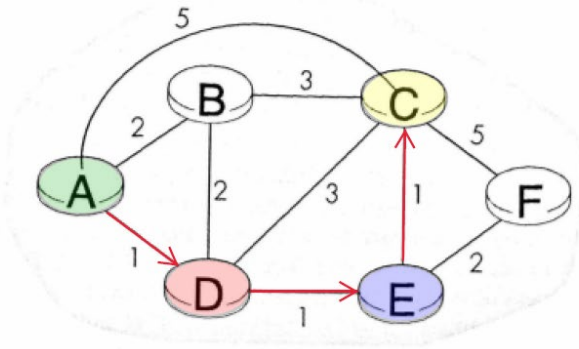
| No. | '821 Patent Claim 2 | The Reference |
|---|---|---|
| 2[preamble] | The method of claim 1, wherein said step of selecting an entity pair further comprises: | The Reference discloses the method of claim 1, wherein said step of selecting an entity pair further comprises.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[c]. |
| 2[a] | selecting a working entity from said entity pair, and | The Reference discloses selecting a working entity from said entity pair.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |
| 2[b] | selecting a protection entity from said entity pair. | The Reference discloses selecting a protection entity from said entity pair.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

| No. | '821 Patent Claim 2 | The Reference |
|---|---|---|
| | | skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 3 | The Reference |
|---|---|---|
| 3 | The method of claim 2, further comprising the step of selecting an active entity from the set consisting of said working entity and said protection entity. | The Reference discloses the method of claim 2, further comprising the step of selecting an active entity from the set consisting of said working entity and said protection entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 4 | The Reference |
|---|---|---|
| 4 | The method of claim 2, wherein selecting an entity pair further comprises minimizing an overall cost function. | The Reference discloses the method of claim 2, wherein selecting an entity pair further comprises minimizing an overall cost function.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references.<br><br>**Kurose discloses**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path. |

| No. | '821 Patent Claim 4 | The Reference |
|---|---|---|
| | | "The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

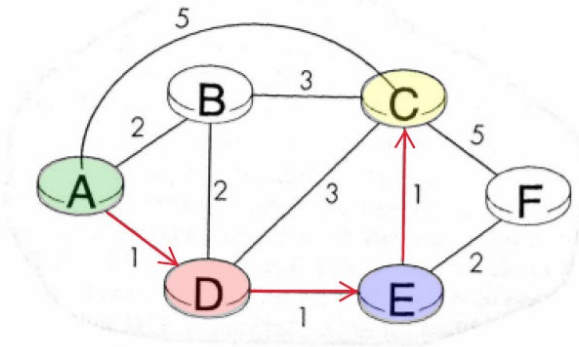
| No. | '821 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>**Figure 4.4** ✦ Abstract model of a network<br><br>✦ the first link in the path is connected to the source<br>✦ the last link in the path is connected to the destination<br>✦ for all $i$, the $i$ and $i-1$st link in the path are connected to the same node<br>✦ for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).<br><br>In Figure 4.4, for example, the least-cost path between nodes *A* (source) and *C* (destination) is along the path *ADEC*. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)<br>As a simple exercise, try finding the least-cost path from nodes *A* to *F*, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from *A* to *F* by examining Figure 4.4, tracing a few routes from *A* to *F*, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between *A* and *F*? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:<br><br>Kurose at 281. |

| No. | '821 Patent Claim 4 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |

| No. | '821 Patent Claim 5 | The Reference |
|---|---|---|
| 5 | The method of claim 4, wherein said overall cost function comprises substantially minimizing a probability of concurrent failure of said protection entity and said working entity. | The Reference discloses the method of claim 4, wherein said overall cost function comprises substantially minimizing a probability of concurrent failure of said protection entity and said working entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

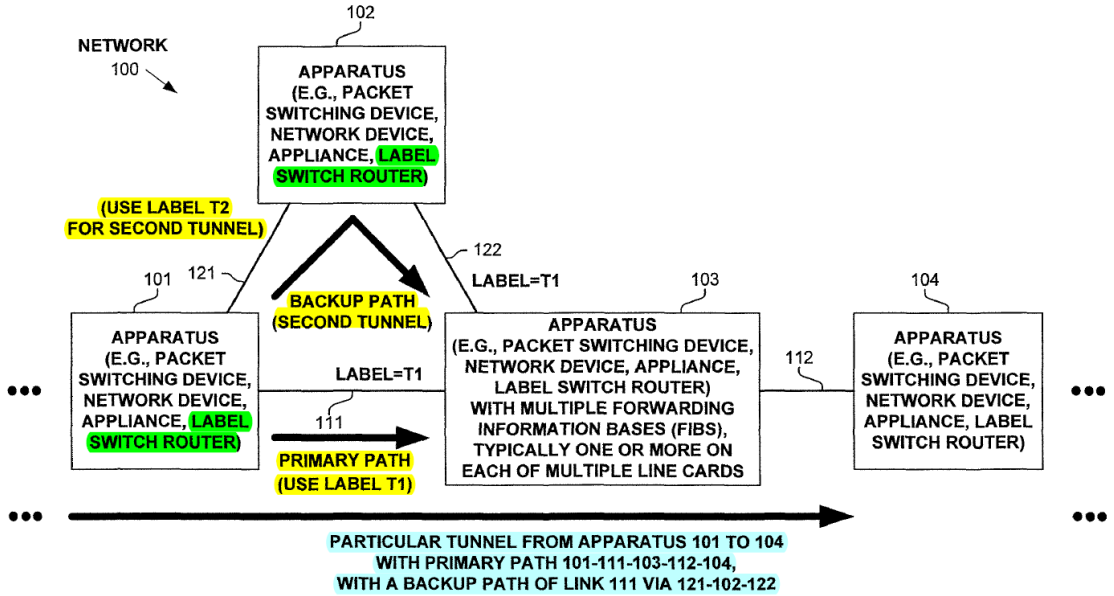
| No. | '821 Patent Claim 6 | The Reference |
|---|---|---|
| 6 | The method of claim 4, wherein said overall cost function comprises a predefined entity cost metric. | The Reference discloses the method of claim 4, wherein said overall cost function comprises a predefined entity cost metric.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references. |

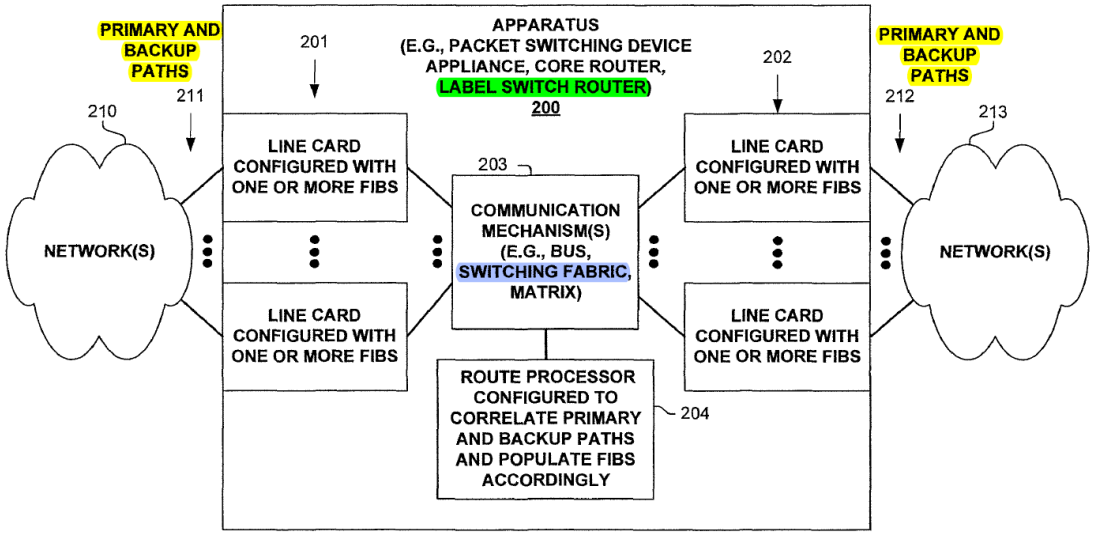
| No. | '821 Patent Claim 6 | The Reference |
|---|---|---|
| | | **Kurose discloses**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path.<br><br>"The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

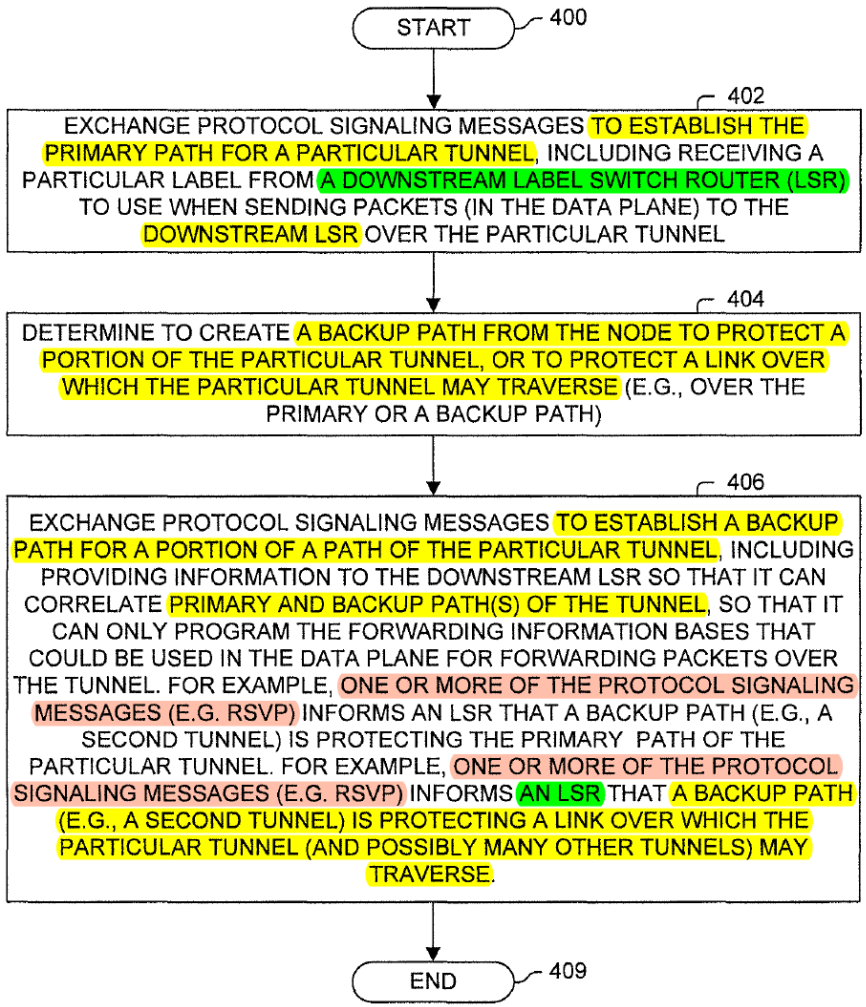
| No. | '821 Patent Claim 6 | The Reference |
|---|---|---|
| | |  <br>**Figure 4.4** ✦ Abstract model of a network<br><br>✦ the first link in the path is connected to the source<br>✦ the last link in the path is connected to the destination<br>✦ for all $i$, the $i$ and $i-1$st link in the path are connected to the same node<br>✦ for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).<br><br>In Figure 4.4, for example, the least-cost path between nodes *A* (source) and *C* (destination) is along the path *ADEC*. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)<br>As a simple exercise, try finding the least-cost path from nodes *A* to *F*, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from *A* to *F* by examining Figure 4.4, tracing a few routes from *A* to *F*, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between *A* and *F*? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:<br><br>Kurose at 281. |

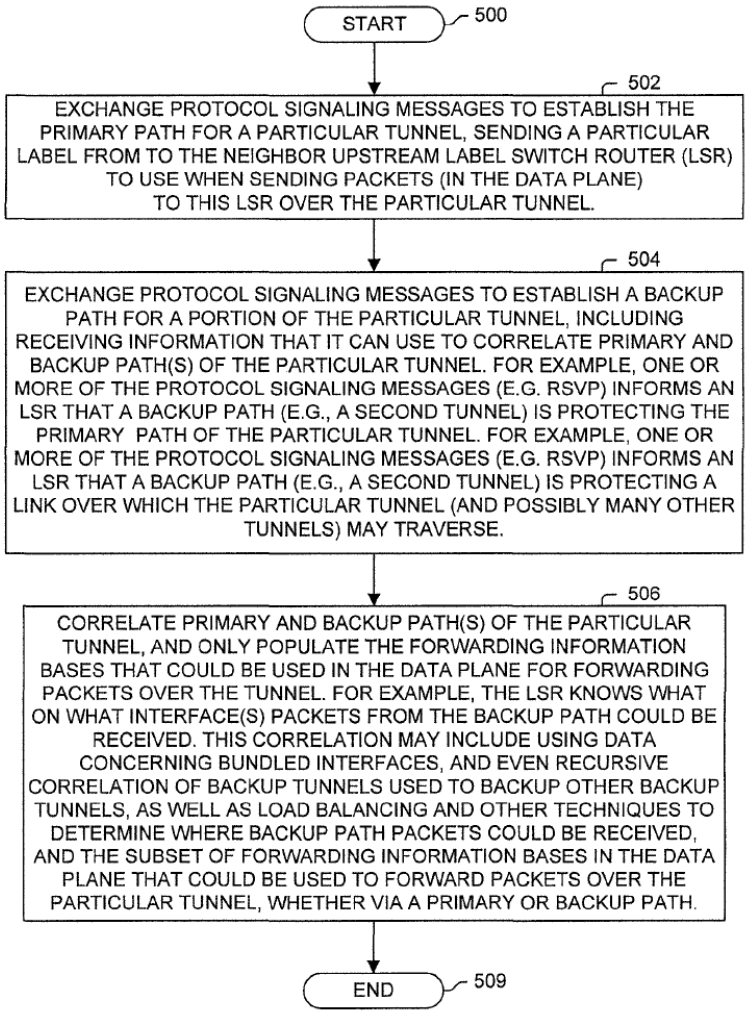
| No. | '821 Patent Claim 6 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| 7 | The method of claim 6, wherein said predefined entity cost metric is selected from the group consisting of interior gateway protocol (IGP) and traffic engineering (TE). | The Reference discloses the method of claim 6, wherein said predefined entity cost metric is selected from the group consisting of interior gateway protocol (IGP) and traffic engineering (TE).<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Cisco created and developed the MPLS and MPLS-TE standards and patented technology based on those standards *before* Orckit utilized such technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Filsfils<br>• Taylor<br>• Vasseur '879<br>• Rustogi<br><br>**Filsfils discloses**:<br>"In one embodiment, forwarding information bases (FIBs) are selectively populated in a packet switch. A packet switching device determines, based on one or more protocol signaling messages, a subset, which is less than all, on which FIBs a lookup operation may be performed for identifying forwarding information for a received particular packet. The packet switching device populates each of these FIBs, but not all of the FIBs of the packet switching device, |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | with forwarding information corresponding to the particular forwarding value. Thus, FIB resources are consumed for only those FIBs which could actually be used, and not all of the FIBs, for forwarding packets in the data plane of the packet switching device, whether these packets are received on a primary or backup path." Filsfils, Abstract.<br>  <br><br>**FIGURE 1**<br>Filsfils, FIG. 1 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIGURE 2<br>Filsfils, FIG. 2 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 4**<br>Filsfils, FIG. 4 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 5**<br>Filsfils, FIG. 5. |

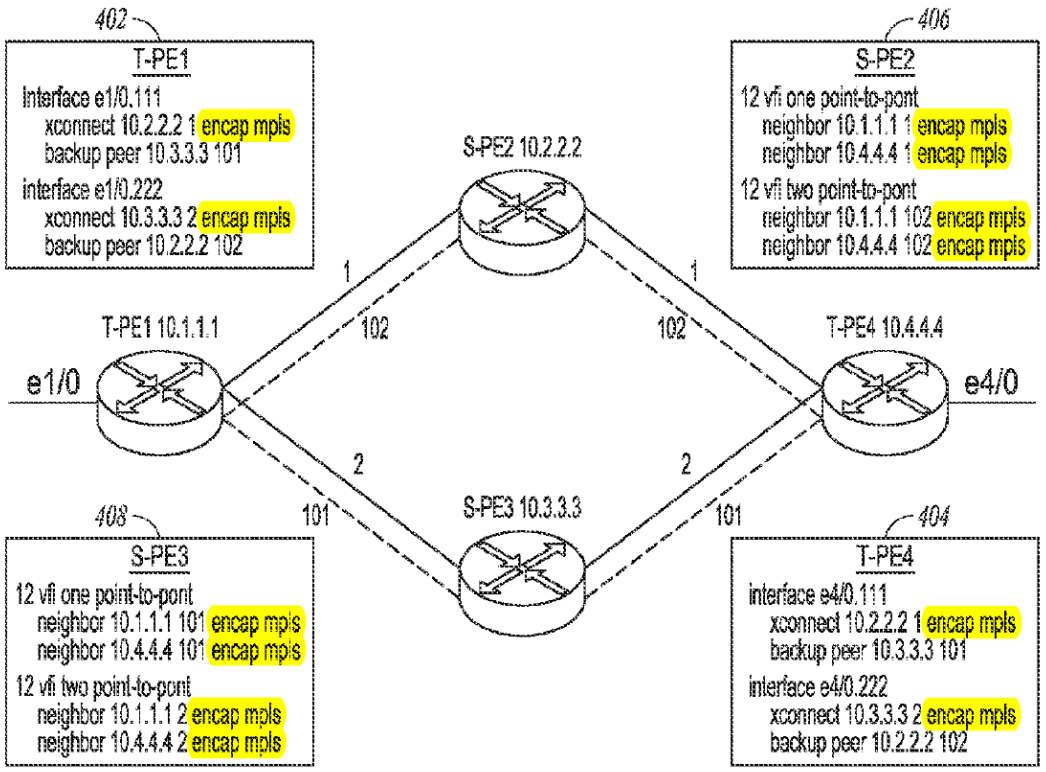
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology, including switching packets on labels especially in the core network using Multiprotocol Label Switching (MPLS)." Filsfils, 1:12-22.<br><br>"Tunnels, such as MPLS-TE (Traffic Engineering) and MPLS-TP (Transport Profile), are paths established through a network in order to transport packets efficiently through a label switched network. Fast Re-Route (FRR) is a technology that allows backup paths to be established in the network, which can be used in case of a problem with a primary path (original primary path or currently used backup path) of the tunnel. RFC 4090, entitled "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," provides an extension of the protocol signaling to establish backup label switched path (LSP) tunnels for local repair of LSP tunnels." Filsfils, 1:23-33.<br><br>"Expressly turning to the figures, FIG. 1 illustrates a network 100 operating according to one embodiment. Shown are four apparatus 101-104 (e.g., packet switching devices such as a label switch router, network device, and/or appliance). For explanation purposes, each of apparatus 101-104 will be referenced as a label switch router (LSR)." Filsfils, 5:41-46.<br><br>"As shown, a particular tunnel is established, using a signaling protocol and exchanging of protocol signaling messages. Note, LSR 101 may, or may not, be an endpoint of the particular tunnel (e.g., LSR 101 may be an intermediate LSR on the path of the particular tunnel). The primary path of the particular tunnel includes spans from LSR 101 via link 111 to LSR 103 and via link 112 to LSR 104. Note, LSR 104 may be an intermediate LSR on the path of the particular tunnel, or an endpoint of the particular tunnel. Further, for this example embodiment, LSR 103 signals LSR 101 to use label T1 at the top of the label stack in the header of a packet sent to it on the particular tunnel." Filsfils, 5:47-58.<br>"A second tunnel from LSR 101 via link 121 to LSR 102 and via link 122 to LSR 103 is similarly configured using a signaling protocol and exchanging of protocol signaling |

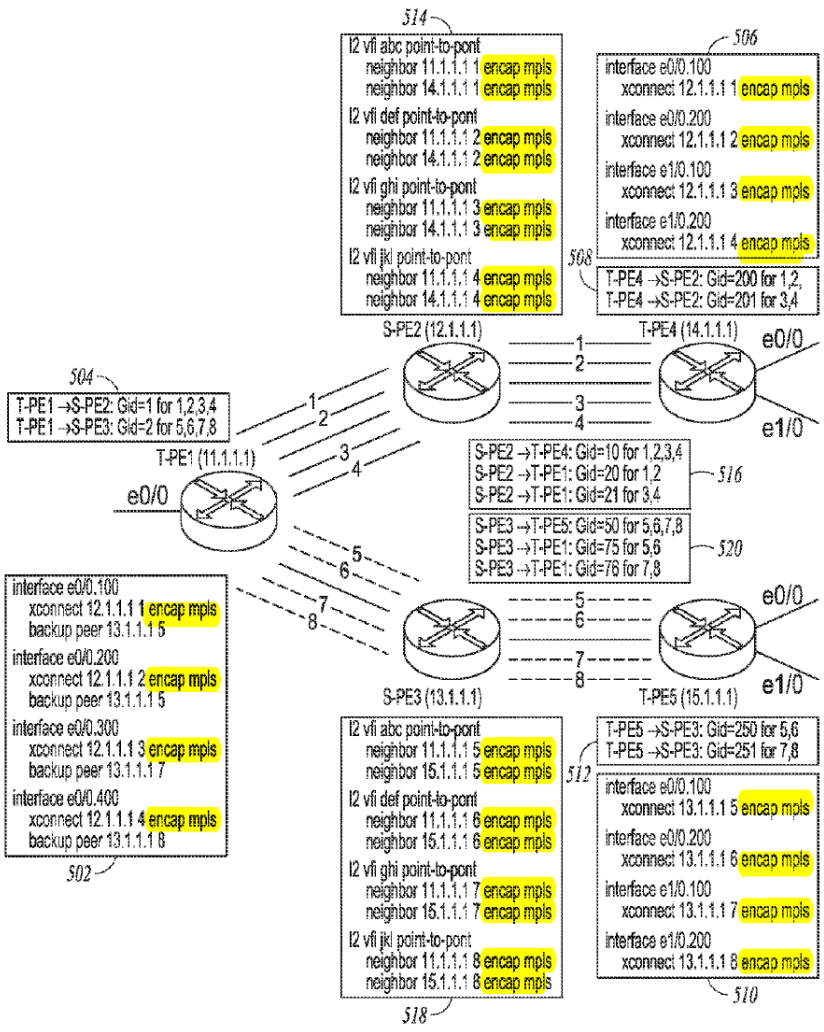
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | messages. For example purposes, LSR 102 signals LSR 101 to use label (T2) at the top of the label stack in the header of a packet sent to it on the second tunnel. In one embodiment, LSR 101 creates the second tunnel in response to determining, or being instructed to, create a backup path to protect link 111 and/or protect all or certain tunnels traversing link 111." Filsfils, 5:59-67.<br><br>"As shown in FIG. 1, link 111 (primary path of the particular tunnel and/or all or certain tunnels traversing link 111) is protected by LSR 101 using the second tunnel (backup path). When sending packets over the particular tunnel over link 111, LSR 101 includes label T1 at the top of the label stack of these packets. If link 111 cannot be used for communicating packets of the particular tunnel, LSR 101 sends packets over the backup path for the particular tunnel by sending packets to LSR 102, with these packets having a label stack including: label T2 followed by label T1. Thus, LSR 102 will receive these packets, pop the top label (T2) from the label stack of each of these packets, identify based on label T2 to send these packets to LSR 103. After popping the top label, the label at the top of the label stack of these packets is T1, which is the same label LSR 103 expects to receive for the particular tunnel. Therefore, these packets received with label T1 at the top of their label stack, are forwarded (after popping label T1 from their label stack) by LSR 103 over the particular tunnel to LSR 104." Filsfils, 6:6-24.<br><br>"One embodiment acquires such additional information by extending Resource Reservation Protocol (RSVP) to provide information which allows a packet switch to correlate primary and backup paths. Thus, a packet switch can use this additional information in determining which of its forwarding information bases (FIBs) could possibly be used in forwarding packets (e.g., in the data plane of the packet switch)." Filsfils, 6:51-57.<br><br>"In providing this additional information to LSR 103, one embodiment communicates an extended RSVP message (including a new or modified RSVP object) or other message to LSR 103 on the second tunnel. This messages designates one or more primary tunnels (e.g., label T1 in our example) and/or a link (e.g., link 111). As LSR 103 knows what interface that it received this message, LSR 103 knows that it must populate forwarding information for these primary tunnels, either specified (e.g., by a label such as T1), or all labels corresponding to |

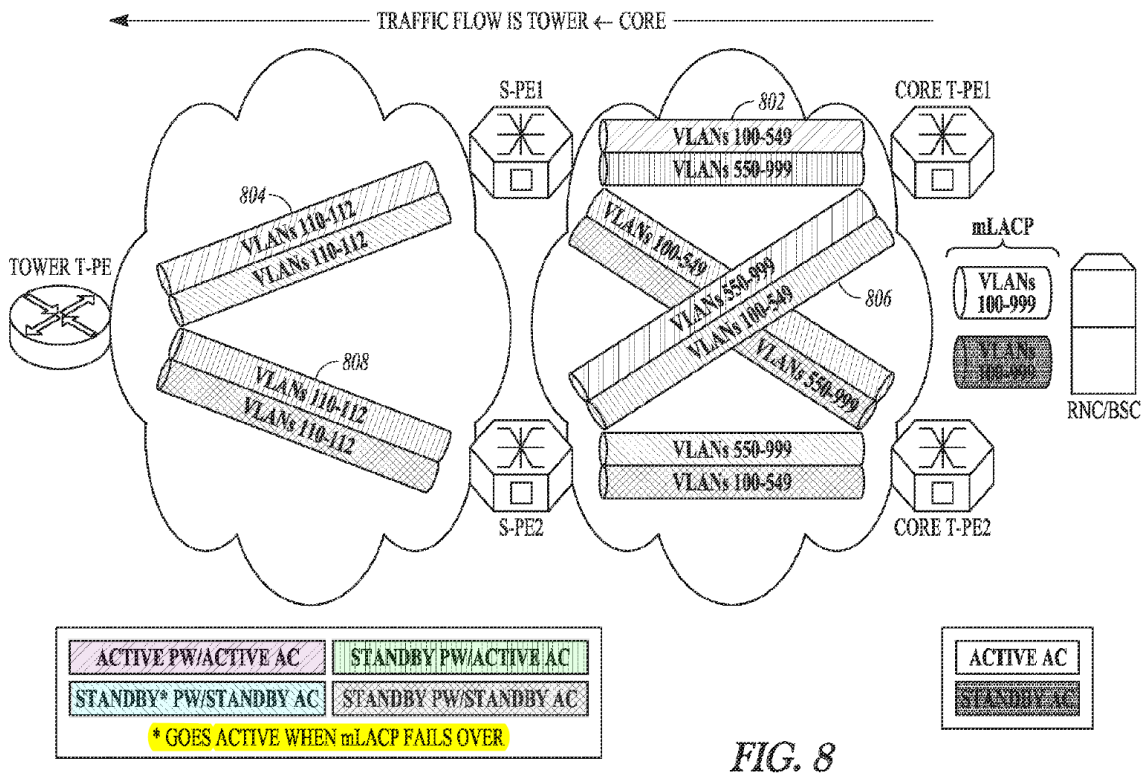
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | tunnels which could be received over link 111. In one embodiment, the extended RSVP or other message communicated to LSR 103 also includes an identification of the backup tunnel (e.g., T2) over which the RSVP or other message is being received, as the identification the tunnel over which a packet is received is often not communicated in a packet (e.g., in the case of Penultimate Hop Popping)." Filsfils, 7:19-34.<br><br>"As shown in FIG. 1, one embodiment includes apparatus 103, which populates less than all of its FIBs with forwarding information for a tunnel (although all FIBs may be populated for certain tunnels). One embodiment includes apparatus 101 and/or 102 which communicates, via a signaling protocol (e.g., an extension of RSVP, or using another protocol), information which allows apparatus 103 to determine the relationship between primary and backup paths, such that apparatus 104 can correlate this primary and backup path information (possibly also correlating backup path of backup path information, and/or bundled interfaces and/or bundled links) to identify a minimum subset of the FIBs that could possibly be used in forwarding packets of particular primary paths (e.g., tunnels)." Filsfils, 7:49-62<br><br>"Turning to FIG. 2, illustrates an apparatus 200 (e.g., packet switching devices such as a label switch router, network device, and/or appliance) operating in one embodiment. As shown, apparatus 200 includes line cards 201, 202 communicatively coupled via communication mechanism(s) 203 (e.g., bus, switching fabric, and/or matrix). Additionally, route processor 204 is configured to correlate primary and backup paths of tunnels, and to populate minimum subsets of FIBs with forwarding information for labels. Again, a minimum subset of FIBs for a particular path or label of the particular path is the set of FIBs that are determined to possibly be used in forwarding packets of a primary path, whether the label is received in a packet over the primary path or over a backup path, and possibly considering backup paths of a backup path and/or the possibly effect of bundled interfaces and/or bundled links." Filsfils, 7:63-8:11.<br><br>"As shown in FIG. 2, apparatus 200 is communicatively coupled via primary and backup paths 211, 212 to networks 210 and 213 (which could be the same network). As illustrated, each of line cards 201, 202 includes one or more FIBs. By correlating on which line card(s) 201, 202 and even within line cards 201, 202 that have multiple FIBs, primary and backup path(s) of tunnels, the number of FIB entries populated in apparatus 200 can typically be reduced, |

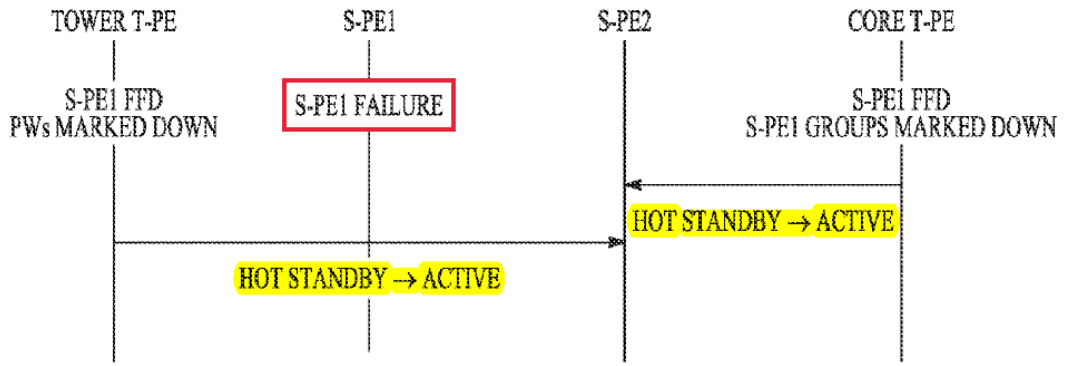
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | possibly significantly saving memory/storage resources and resources used to populate the FIBs." Filsfils, 8:12-21.<br><br>"FIG. 4 illustrates a process performed in one embodiment. Processing begins with process block 400. In process block 402, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes receiving a particular label for a downstream LSR to use when sending packets to the downstream LSR over the particular tunnel." Filsfils, 8:61-67.<br><br>"In process block 404, a determination is made to create a backup path from the node (e.g. the node performing these operations). This backup path may be used to protect one or more particular tunnels, and/or may be used to protect a link which is used to carry packet traffic of one or more tunnels." Filsfils, 9:1-5.<br><br>"In process block 406, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including providing information to the downstream LSR so that the downstream LSR can correlate primary and backup path(s) of the particular tunnel and substantially only program the FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:6-23.<br><br>"FIG. 5 illustrates a process performed in one embodiment. Processing begins with process block 500. In process block 502, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes sending a particular label for an upstream LSR to use when sending packets over the particular tunnel to this apparatus (e.g., an LSR performing these operations)." Filsfils, 9:26-32. |

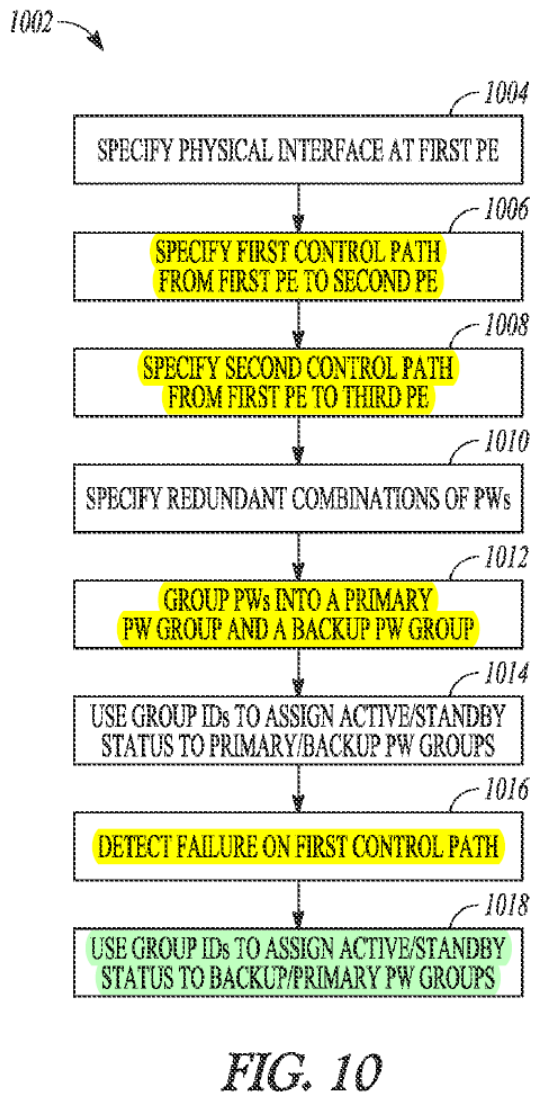
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "In process block 504, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including receiving information that the LSR can use to correlate primary and backup path(s) of the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:33-47.<br><br>"In process block 506, the primary and backup path(s) of the particular tunnel are correlated to identify the set of FIBs that could possibly be used in forwarding packets of the particular tunnel. Substantially only those FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel (either through a primary or backup path) are populated with the forwarding information (e.g., an entry corresponding to the label it advertised to use for the particular tunnel) for the particular tunnel. For example, the LSR knows what on what interface(s) packets from the backup path could be received. This correlation may include using data concerning bundled interfaces, and even recursive correlation of backup tunnels used to backup other backup tunnels, as well as load balancing and other techniques to determine where backup path packets could be received, and the subset of forwarding information bases in the data plane that could be used to forward packets over the tunnel, whether via a primary or backup path." Filsfils, 9:48-65.<br><br>**Taylor discloses**:<br>"Grouping pseudowires based on hardware interfaces and configured control paths enables improved pseudowire failover performance. Signaling status changes (e.g., from standby to active status) is facilitated by using group IDs for the pseudowire groups, thereby enabling improved failover performance when there is disruption in the network." Taylor, Abstract. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Taylor, FIG. 4 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>*FIG. 5*<br>Taylor, FIG. 5 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br>Taylor, FIG. 8 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Taylor, FIG. 9 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Taylor, FIG. 10 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "The present disclosure relates generally to communication networks and more particularly to pseudowire configurations in communication networks." Taylor, 1:8-10.<br><br>"Virtual Private Network (VPN) services provide secure network connections between different locations. A company, for example, can use a VPN to provide secure connections between geographically dispersed sites that need to access the corporate network so that each customer edge (CE) end point or node can communicate directly and independently with all other CE nodes. Different types of VPNs have been classified by the network layer used to establish the connection between the customer and provider network. For example, Virtual Private LAN Service (VPLS) is an architecture that delivers a multipoint Layer 2 VPN (L2VPN) service that in all respects emulates an Ethernet Local Area Network (LAN) across a wide metropolitan geographic area. All services in a VPLS appear to be on the same LAN, regardless of location. In other words, with VPLS, customers can communicate as if they were connected via a private Ethernet segment, i.e., multipoint Ethernet LAN services." Taylor, 1:12-28.<br><br>"In this context, each CE device at a customer site is connected to the service provider network at a provider edge PE) device by an Attachment Circuit (AC) that provides the customer connection to a service provider network, that is, the connection between a CE node and its associated PE node. Within the provider network, each PE device includes a Virtual Switch Instance (VSI) that emulates an Ethernet bridge (i.e., switch) function in terms of Media Access Control (MAC) address learning and forwarding in order to facilitate the provisioning of a multipoint L2VPN. A pseudowire (PW) is a virtual connection between two PE devices that connect two attachment circuits. In the context of the VPLS service, a pseudowire can be thought of as a point-to-point virtual link for each offered service between a pair of VSIs. Therefore, if each VSI can be thought of as a virtual Ethernet switch for a given customer service instance, then each pseudowire can be thought of as a virtual link connecting these virtual switches to each other over a Packet Switched Network (PSN) for that service instance." Taylor, 1:29-47. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Since the failure of pseudowires obviously degrades network performance, some effort has been directed towards adding system redundancies including redundant pseudowires. However, the presence of redundant pseudowires alone is insufficient to improve overall failover performance, that is, the ability to switch over automatically to a redundant or backup system. Thus, there is a need for improved methods for managing pseudowires to facilitate pseudowire switching and enable improved failover performance." Taylor, 1:48-56.<br><br>"FIG. 4 shows details for PW connectivity in an exemplary network for an example embodiment." Taylor, 1:66-67.<br><br>"FIG. 5 shows details for PW grouping for an example embodiment." Taylor, 2:1-2.<br><br>"FIG. 8 shows an example network including redundant PW connections for an example embodiment." Taylor, 2:9-10.<br><br>"FIG. 9 shows an example sequence diagram for a failure mode related to the embodiment shown in FIG. 8." Taylor, 2:11-12.<br><br>"FIG. 10 shows a flowchart that illustrates a method of providing improved PW grouping according to an example embodiment." Taylor, 2:13-15.<br><br>"According to one embodiment, a method of providing improved pseudowire performance includes specifying a physical interface at a first PE node in a network, a first control path from the first PE node to a second PE node in the network, and a second control path from the first PE node to a third PE node in the network. With these specifications, the method then includes specifying redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node, and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Then these pseudowires can be grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. Group identifiers for the primary PW group and the backup PW group can then be used to assign an |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths. The method may then include detecting a failure on the first control path, and in response to the detected failure, using the group identifiers to assign the active status to the backup pseudowires and the standby status to the primary pseudowires." Taylor, 2:27-52.<br><br>"Pseudowires are used in pseudowire emulation edge-to-edge to provide a Layer 2 Virtual Private Network (L2VPN) connection. When large numbers (e.g., 4,000-6,000) of pseudowires are aggregated together on a single router, failure performance tends to be linear or O(n) where n is the number of pseudowires. While O(n) performance may be acceptable for small numbers of pseudowires, the effect on network outages can be increasingly undesirable as the number of pseudowires increases." Taylor, 2:54-62.<br><br>"For example, a cell-site router will typically start an approximately 2-minute procedure if contact with its controller, which is reached via a pseudowire, is lost for more than some threshold amount (e.g., between approximately 0.75 and 1.75 seconds in some cases). This can be a major impediment to the scalability of pseudowire deployments. These issues have become increasingly relevant as providers of Multiservice Broadband Networks (MBNs) are rapidly replacing or augmenting their traditional Synchronous Optical Networking (SONET) equipment with cheaper Ethernet equipment in the evolution towards a 4G (i.e., 4$^{th}$ generation) network." Taylor, 2:63-3:7.<br><br>"One aspect of a solution to the problem of pseudowire failure is the deployment of redundant pseudowires. For example, redundant pseudowires have been used in the context of Multiprotocol Label Switching (MPLS) networks, which use a Label Distribution Protocol (LDP) to manage labels for forwarding traffic between routers. In this context, general requirements for redundancy schemes have been developed so that duplicate pseudowires are available when a given pseudowire fails (e.g., by using active/standby status indicators). In addition, more specific implementations for redundant pseudowires have also been developed." Taylor, 3:8-18. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 1 shows a reference network model 102 with applications to example embodiments disclosed herein. The reference network model 102 includes an aggregation network 104 of PE nodes and a distribution network 106 of PE nodes between a radio network controller (RNC) (or base station controller (BSC)) 108 on the core side of the model 102 and a radio tower 111 on the tower side of the model 102. Switching provider edge nodes S-PE1 and S-PE2 connect the two networks 104, 106. On the core side, two core terminating provider edges T-PE1 and T-PE2 connect to the RNC/BSC 108 through attachment circuits 110, 112. On the tower side, one tower terminating provider edge T-PE connects to the radio tower 111 through an attachment circuit 114." Taylor, 3:19-31.<br><br>"Additionally as noted in FIG. 1, peer-PE monitoring is carried out within each network 104, 106. That is, there is peer-PE monitoring between provider edges that share a segment, for example, by multi-hop bidirectional forwarding detection (BFD). Alternatively, peer monitoring can be accomplished by other means (e.g., MPLS-TP (Transport Protocol) keep-olives). This peer-PE monitoring is used to provide the mechanism for fast failure detection. Once a failure is detected, the network can react by "rerouting" the failed pseudowires to pre-provisioned backup paths and thus provide a minimal disruption in service to the end-user. This rerouting can be accomplished by LDP signaling between provider edges." Taylor, 3:32-44.<br><br>"The reference network model 102 may be considered as part of a larger hub-and-spoke model as shown in FIG. 2. A hub-and-spoke distribution model 202 includes a core network 204, distribution networks 206, and aggregation networks 208. Network elements including distribution nodes, aggregation nodes, and towers are also shown with nominal count values (e.g., 30 distribution nodes between the core network 204 and a distribution network 206). In this model 202, tower T-PEs are the spokes white core-PEs constitute the hub. Dozens to hundreds of tower T-PEs connect to a few S-PEs; these S-PEs are quite similar to ASBRs as they act as forwarders between the two distinct MPLS domains, providing isolation and, in the case of mobility, aggregation services. Typically, several aggregation networks 208 are connected to a single distribution network 206, eventually connecting the tower with the core router that connects the tower's ACs to the RNC/BSC. There are typically several distribution |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | networks in a Radio Access Network (RAN) connected to the service provider's core Internet Protocol (IP) network." Taylor, 4:11-30.<br><br>"With reference to FIG. 1, FIG. 3 shows a variety of failure modes encountered in the reference network model 102. Failure 302 of communications between tower T-PE and the S-PE can be detected via peer monitoring when both the S-PE and the T-PE are still active/alive. For example, this failure may be due to a toss of connectivity when the BFD session goes down. Failure 304 of S-PE1 can be due to a hardware failure, power outage, or the lack of BED-session maintenance capability (e.g., if the BFD-session hello timers cannot be serviced for the prescribed period of time). Failure 306 of communications between S-PE1 and core T-PE1 can be detected via peer monitoring when both S-PE1 and core T-PE1 are still active/alive. Failure 308 at core T-PE1 can be due to a hardware failure, power outage, or the lack of BFD-session maintenance capability." Taylor, 4:46-60.<br><br>"Pseudowire connectivity is further illustrated in FIG. 4 where the illustrated network includes four nodes: T-PE1 (10.1.1.1), S-PE2 (10.2.2.2), S-PE3 (10.3.3.3), and T-PEA (10.4.4.4). For the terminating nodes T-PE1 and T-PE4, specifications for VLANs (virtual Local Area Networks) connections (i.e., pseudowires) are shown using the Internet Operating System Command Line Interface (IOS CLI). The specification 402 for T-PE1 defines two VLANS as primary/backup combinations of virtual circuits for the network. The first three lines of the specification 402 define "VLAN 111" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 111" in the first line. The second line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE2 (10.12.2) with a virtual circuit Identification (VCID) set as VCID=1, and the third line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=101. The next three lines of the specification 402 define "VLAN 222" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 222" in the fourth line. The fifth line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=2, and the sixth line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE2 (10.2.2.2) with VCID=102. These virtual circuits, VCID=1, VCID=2, VCID=101 and VCID=102 are shown in the figure between T-PE1 and the S-PEs with a solid line for the primary circuits VCID=1 |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | and VCID=2 and a dashed line for the backup circuits VCID=101 and VCID=102." Taylor, 4:61-5:23.<br><br>"With respect to T-PE1 in FIG. 4, although "VLAN 111" and "VLAN 222" share the same hardware port, they do not share the same "control path disposition." That is, "VLAN 111" is primary to S-PE2 (VCID=1) and standby to S-PE3 (VCID=101), white "VLAN 222" has an opposite configuration since it is primary to S-PE3 (VCID=2) and standby to S-PE2 (VCID=102). As discussed below, certain embodiments group pseudowires according to "control path disposition" (e.g., xconnect configuration as well as the hardware interface in order to improve failover performance. That is, to deal with both hardware port failures and switching path failures, the grouping criteria also considers the cross connects. In this case, on T-PE1 as well as T-PE4, there would exist two groups: one for active to S-PE2 and standby to S-PE3 and another for active to S-PE3 and standby to S-PE2 (i.e., the inverse configuration)." Taylor, 5:61-6:9.<br><br>"First, local connectivity is characterized by local group identifications (Group-IDs), which depend on whether the allocation is done at a T-PE or S-PE. FIG. 5 shows an embodiment that illustrates an allocation of local group IDs in a network including terminating nodes T-PE1 (11.1.1.1), T-PE4 (14.1.1.1), and T-PE5 (15.1.1.1) and switching nodes S-PE2 (12.1.1.1) and S-PE3 (13.1.1.1). The specification 502 for T-PE1 determines corresponding local group IDs 504 based on the hardware interface and the control path. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, and the Group-ID=2 corresponds to VCID=5, VCID=6, VCID=7, and VCID=8." Taylor, 6:10-21.<br><br>"Local group IDs are maintained in a database so that pseudowire redundancy is also maintained. First, in a case without pseudowire redundancy, all the xconnect configurations from the same physical interface to the same peer are assigned the same local group ID. So, for example, in Ethernet cases all xconnect configurations under sub-interfaces of the same physical interface to the same peer will be assigned the same local group ID (e.g., e0/0 and e0/1 are sub-interfaces of e0). FIG. 6 shows a database representation for T-PE4 from FIG. 5. From the root node 602 for T-PE4, there is a first interface node 604 for e0/1 and a second interface node 606 for e1/0. The first interface node 604 is configured towards a single peer |

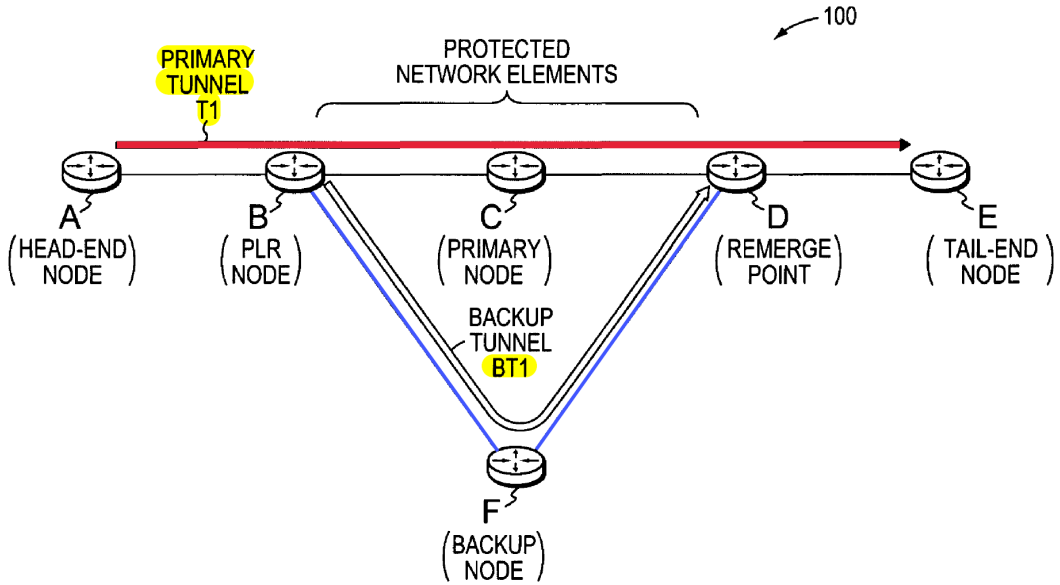
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | node (12.1.1.1) 608 and is thus assigned a single local group ID (Group-ID=200) 610. Similarly, the second interface node 606 is configured towards a single peer node (12.1.1.1) 612 and is thus assigned a single local group ID (Group-ID=201) 614. In this case from the assignment of local group IDs 508 in FIG. 5, Group-ID=200 corresponds to VCID=1 and VCID=2, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). Both pseudowires (VCID=1 and VCID=2) are assigned the same local group ID (Group-ID=200) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly, from the assignment of local group IDs 508 in FIG. 5, T-PE4 has VCID=3 and VCID=4 under the physical interface e1/0 going to the same peer S-PE2 (12.1.1.1), and the local Group ID (Group-ID=201) is assigned to these VCs." Taylor, 6:48-7:8.<br><br>"For the pseudowire redundancy case, a separate redundancy-group database is maintained by the xconnect application. This redundancy-group database contains the peer IDs in the group and the local group IDs advertised to them. This is needed to maintain a 1:1 mapping between the primary pseudowires and their corresponding backup pseudowires. FIG. 7 shows a database representation for T-PE1 from FIG. 5. From the root node 702 for T-PE1, there is an interface node 704 for e0/0 and a redundancy group node 706 that shows connections for configurations to a first peer node (12.1.1.1) 708, which is assigned a local group ID (Group-ID=1) 710, and a second peer node (13.1.1.1) 712, which is assigned a local group ID (Group-ID=2) 714. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). These pseudowires are assigned the same group ID (Group-ID=1) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly from the assignment of local group IDs 504 in FIG. 5, T-PE1 has VCID=5, VCID=6, VCID=7, and VCID=8 under the physical interface e0/0 going to another peer S-PE3 (13.1.1.1), and the local group ID (Group-ID=2) is assigned to these VCs. In this case these local group IDs are organized as a redundancy group 706." Taylor, 7:9-35. |

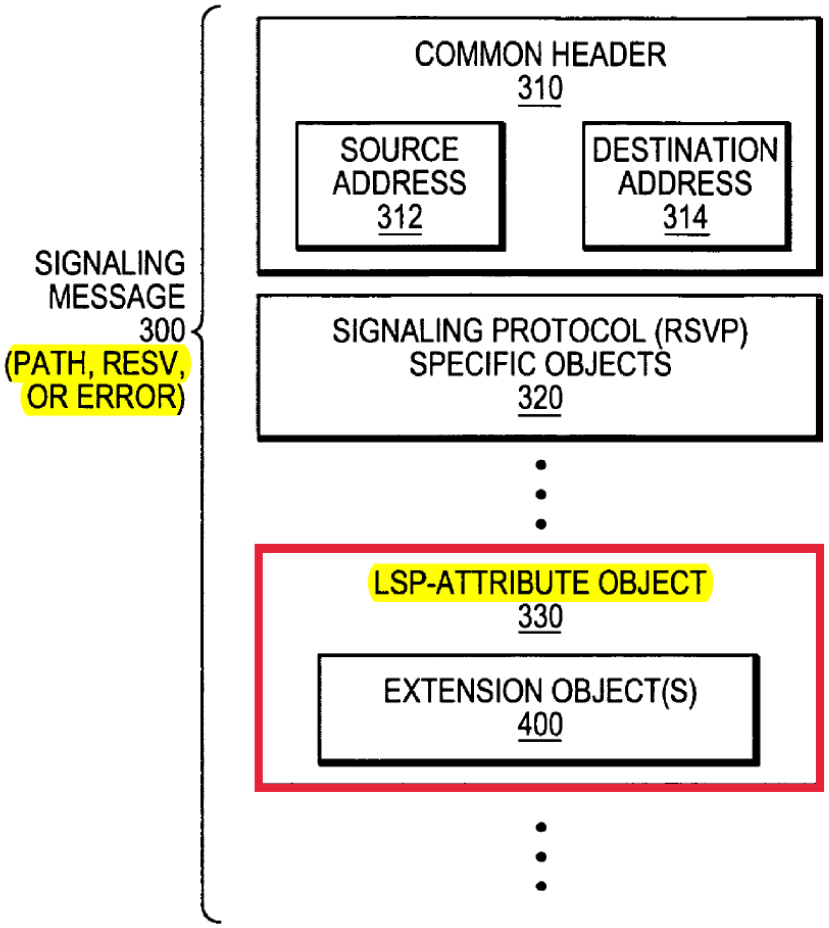
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Pseudowire grouping allows multiple pseudowires to be aggregated together when signaling either label withdrawals or status changes between segment end-point provider edges. This signaling can be carried out through LDP grouping TLV (Type Length Value). For example, when a PE node fails (e.g., failure 304 of S-PE1), aggregating the inter-segment PE signaling using the grouping TLV can provide significant scaling advantages. This allows all pseudowires sharing a physical port and PW configuration (e.g., xconnect configuration) to be signaled en masse between segment-adjacent provider edges." Taylor, 7:36-46.<br><br>"FIG. 8 shows an example based on FIG. 1 where VLAN ACs are shown as grouped by both port/HW-interface and pseudowire-class. The grouping criterion allows all "similar" pseudowires to be signaled together: All the grouped pseudowires share the same port and next-hop provider edge. Additionally, the figure contains many pseudowires, each grouped into a shaded tube. For example, the tube labeled "VLANs 100-549" contains 450 pseudowires grouped together. This figure depicts an incoming Ethernet comprised of 900 VLANs being segmented in two with 450 VLANs (100-549) active to S-PE1 while the other half of the VLANs (550-999) being active to S-PE2. This might be considered a type of manual load balancing. Furthermore, the aggregation network is only showing a single tower and the VLANs associated with it; other VLAN destinations are not shown in the figure." Taylor, 7:47-62.<br><br>"'VLANs 110-112' are active along a first pseudowire path 802 from Core T-PE1 to S-PE1 and a second pseudowire path 804 from S-PE1 to Tower T-PE. When a failure occurs at S-PE1 (e.g., as the switching node failure 304 shown in FIG. 3), then the standby pseudowires become active for 'VLANs 110-112' along a first pseudowire path 806 from Core T-PE1 to S-PE2 and a second pseudowire path 808 from S-PE2 to Tower T-PE." Taylor, 7:63-8:3.<br><br>"The standby pseudowires in FIG. 8 can be configured as HSPWs, a configuration that enables ACs to quickly failover to pre-provisioned pseudowires that are in active state but set to not-forwarding. Then when a failure occurs, switching over to these pre-provisioned HSPWs occurs quickly by switching from not-forwarding status to forwarding status." Taylor, 8:4-9. |

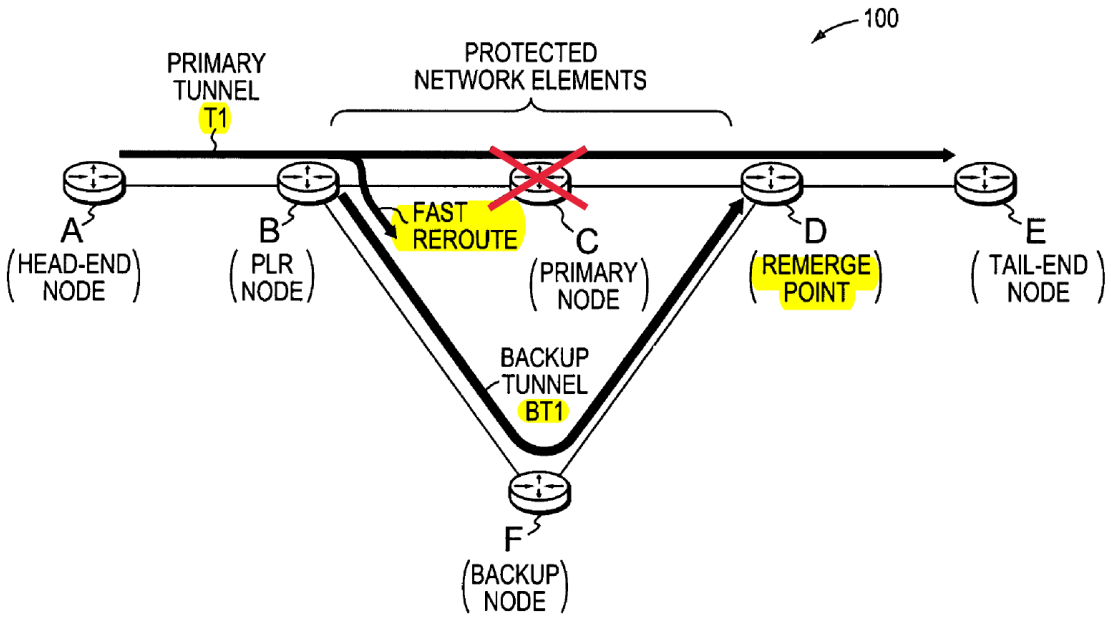
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 9 shows a UML (Unified Modeling Language) sequence diagram of signaling events related to the failover procedure illustrated in FIG. 8 for a switching node failure 304. The Tower T-PE and the Core T-PE independently detect a failure at S-PE1 (e.g., BFD forwarding with LDP signaling), and then mark down the status of the currently active pseudowires routed through S-PE1 and mark up the status of the standby pseudowires routed through S-PW2. Other failure modes shown in FIG. 3 can be handled similarly." Taylor, 8:10-18.<br><br>"In general, it is desirable for MPLS-based. Ethernet networks to react quickly to failures, so proactive detection mechanisms are employed in order to pick up system failures quickly. All proactive monitoring is typically done between PE peers on a single MPLS network. These provider edges on the edges of the MPLS network act similarly to an Autonomous System Boundary Router (ASBR). As a result, related embodiments detect control path failures, which may not be the same as pseudowire data path failures. That is, the data packets and control packets may take different paths between provider edges in a MPLS network although typically these paths are coincident. Thus, when the control and data paths are not coincident, if the control path fails, then all pseudowires utilizing the control path are marked as failed. As a corollary, if the data path fails and the control path remains healthy, then failure will not be detected from monitoring the control path." Taylor, 8:19-35.<br><br>"A failure of a monitored provider edge initiates a switchover of all active pseudowires using the failing provider edge to their configured HSPWs (if they exist). Grouping can greatly reduce the number of messages needed between provider edges (Inter-PE Aggregation) and within a single provider edge (Intra-PE Aggregation). Furthermore, the MPLS network itself may be internally resilient deploying technologies such as, but not limited to, MPLS-TE (MPLS Traffic Engineering) and ERR (Fast Reroute). The paths across the MPLS network may recover quickly and might not trip the fault-monitoring systems." Taylor, 8:36-46.<br><br>"With reference to the above discussion, FIG. 10 shows a method 1002 of providing improved PW grouping according to an example embodiment. In a first operation 1004 of the method 1002, a physical interface is specified at a first PE node in a network. In a second operation 1006, a first control path is specified from the first PE node to a second PE node in the network. In a third operation 1008, a second control path is specified from the first PE node to a third |

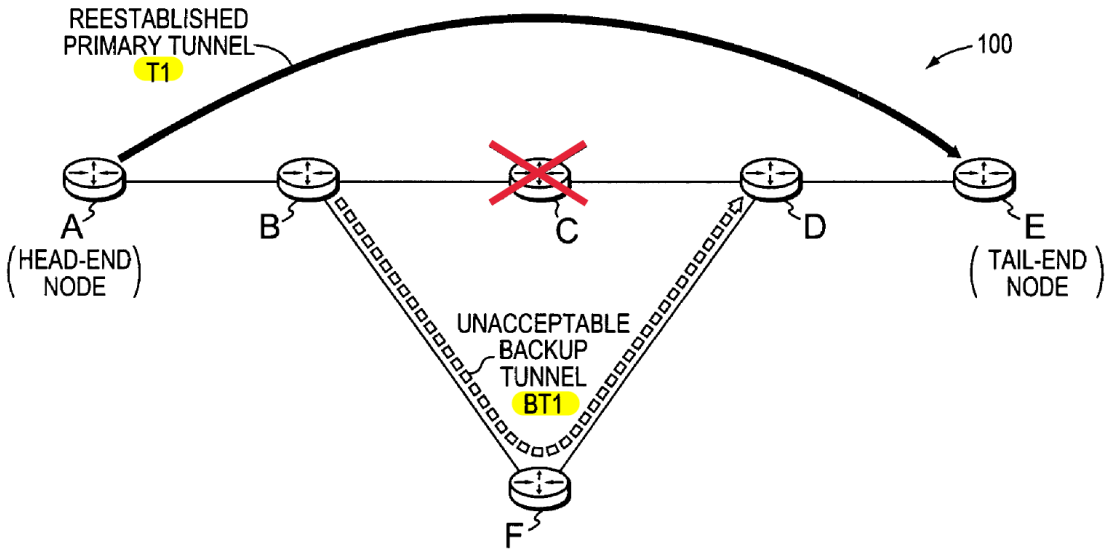
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | PE node in the network. These control paths related to a common physical interface can be used to characterize redundant pairs of pseudowires." Taylor, 8:48-58.<br><br>"In a fourth operation 1010, redundant combinations of pseudowires are specified, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Although a redundant combination may relate a single backup pseudowire to a given primary pseudowire, in some cases multiple backup pseudowires will be related to a given primary pseudowire for increased redundancy. In a fifth operation 1012, these pseudowires are grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. In a sixth operation 1014, group identifiers for the primary PW group and the backup PW group are used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 8:59-9:11.<br><br>"In an optional seventh operation 1016, a failure may be detected on the first control path, and in an optional eighth operation 1018, in response to the detected failure, the group identifiers may be used to assign the active status to the backup pseudowires and the standby status to the primary pseudowires. For example, the failure on the first control path may be detected by using BED packet streams between PE nodes of the network. Then the detected failure can be signaled to PE nodes in the network by sending LDP status updates between PE nodes in the network. Then, after receiving the failure detection signals, the group identifiers can be used again to assign the active status to the backup pseudowires and the standby status to the primary pseudowires by sending LDP status updates between PE nodes in the network." Taylor, 9:12-26.<br><br>"Typically the network in is an MPLS network and the PE nodes are routers that provide network services to connected CE nodes of a customer network. In general, each control path is an Internet Protocol (IP) routing path between PE nodes in the network and each data path is a label switched path (LSP) between PE nodes in the network." Taylor, 9:27-32. |

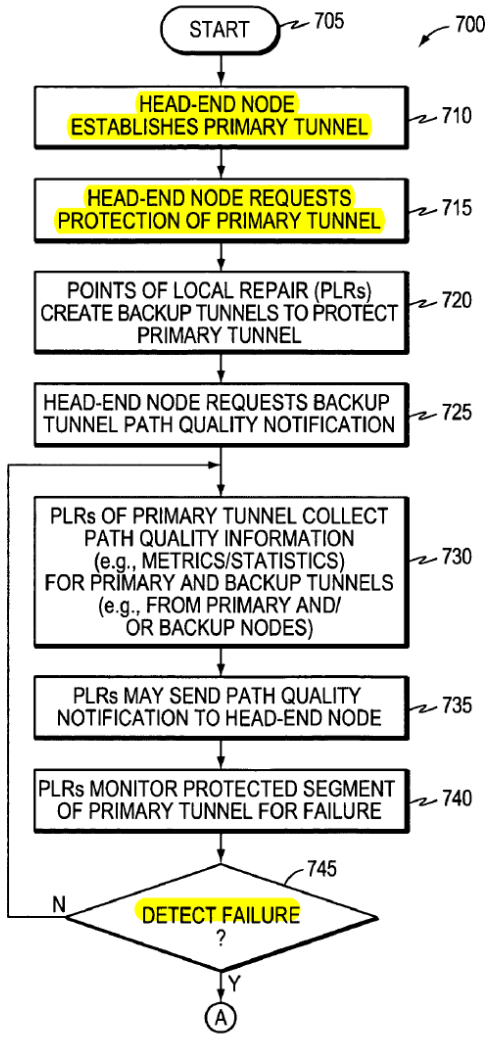
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 11 shows a schematic representation of an apparatus 1102, in accordance with an example embodiment. For example, the apparatus 1102 may be used to implement the method 1002 of providing improved pseudowire grouping as described above with reference to FIG. 10. The apparatus 1102 is shown to include a processing system 1104 that may be implemented on a server, client, or other processing device that includes an operating system 1106 for executing software instructions." Taylor, 10:2-10.<br><br>"In accordance with an example embodiment, the apparatus 1102 includes a PW management module 1108 that includes a first specification module 1110, a second specification module 1112, third specification module 1114, a fourth specification module 1116, a grouping module 1118, and an assignment module 1120. The first specification module 1110 operates to specify a physical interface at a first PE node in a network. The second specification module 1112 operates to specify a first control path from the first PE node to a second PE node in the network. The third specification module 1114 operates to specify a second control path from the first PE node to a third PE node in the network. The fourth specification module 1116 operates to specify redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node." Taylor, 10:11-29.<br><br>"The grouping module 1118 operates to group the pseudowires into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. The assignment module 1120 operates to use group identifiers for the PW groups to assign an active status to the primary pseudowires and a standby status to the backup pseudowires. The active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 10:30-38. |

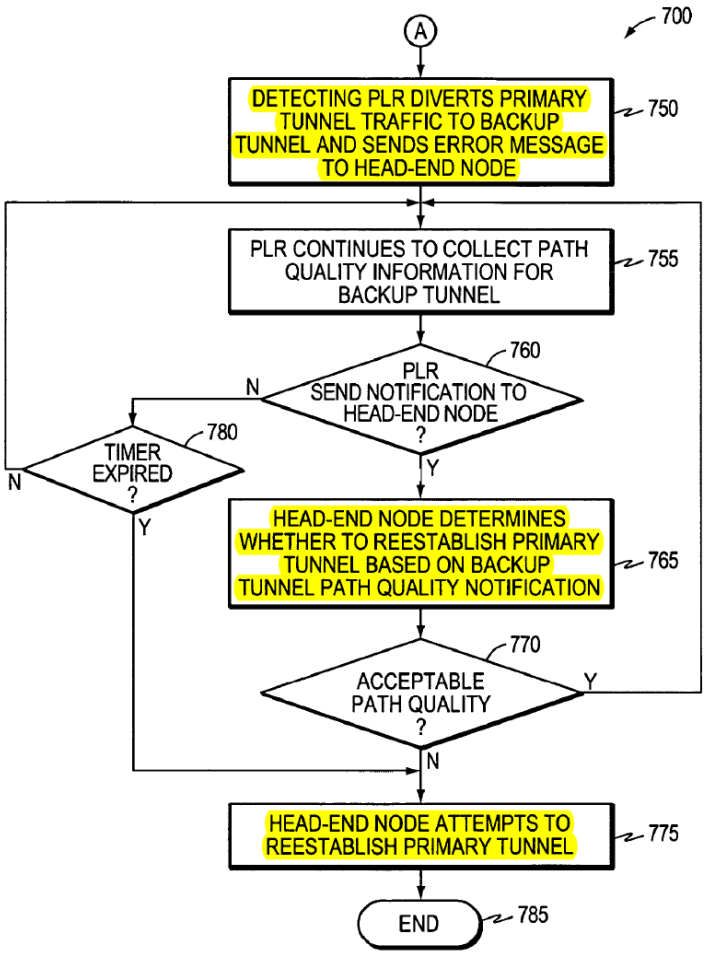
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | **Vasseur '879 discloses**:<br>"A technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, Abstract.<br>  <br><br>FIG. 1<br>Vasseur '879, FIG. 1 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Vasseur '879, FIG. 3 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Vasseur '879, FIG. 5 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br>Vasseur '879, FIG. 6 (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 7A<br>Vasseur '879, FIG. 7A (annotated). |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 7B<br><br>Vasseur '879, FIG. 7B. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "The present invention relates to computer networks and more particularly to dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 1:10-14.<br><br>"Since management of interconnected computer networks can prove burdensome, smaller groups of computer networks may be maintained as routing domains or autonomous systems. The networks within an autonomous system (AS) are typically coupled together by conventional "intradomain" routers configured to execute intradomain routing protocols, and are generally subject to a common authority. To improve routing scalability, a service provider (e.g., an ISP) may divide an AS into multiple "areas." It may be desirable, however, to increase the number of nodes capable of exchanging data; in this case, interdomain routers executing interdomain routing protocols are used to interconnect nodes of the various ASes. Moreover, it may be desirable to interconnect various ASes that operate under different administrative domains. As used herein, an AS or an area is generally referred to as a "domain," and a router that interconnects different domains together is generally referred to as a 'border router.'" Vasseur '879, 1:40-56.<br><br>"An example of an interdomain routing protocol is the Border Gateway Protocol version 4 (BGP), which performs routing between domains (ASes) by exchanging routing and reachability information among neighboring interdomain routers of the systems. An adjacency is a relationship formed between selected neighboring (peer) routers for the purpose of exchanging routing information messages and abstracting the network topology. The routing information exchanged by BGP peer routers typically includes destination address prefixes, i.e., the portions of destination addresses used by the routing protocol to render routing ("next hop") decisions. Examples of such destination addresses include IP version 4 (IPv4) and version 6 (IPv6) addresses. BGP generally operates over a reliable transport protocol, such as TCP, to establish a TCP connection/session. The BGP protocol is well known and generally described in Request for Comments (RFC) 1771, entitled A Border Gateway Protocol 4 (BGP-4), published March 1995." Vasseur '879, 1:57-2:7. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Examples of an intradomain routing protocol, or an interior gateway protocol (IGP), are the Open Shortest Path First (OSPF) routing protocol and the Intermediate-System-to-Intermediate-System (IS-IS) routing protocol. The OSPF and IS-IS protocols are based on link-state technology and, therefore, are commonly referred to as link-state routing protocols. Link-state protocols define the manner with which routing information and network-topology information are exchanged and processed in a domain. This information is generally directed to an intradomain router's local state (e.g., the router's usable interfaces and reachable neighbors or adjacencies). The OSPF protocol is described in RFC 2328, entitled OSPF Version 2, dated April 1998 and the IS-IS protocol used in the context of IP is described in RFC 1195, entitled Use of OSI IS-IS for routing in TCP/IP and Dual Environments, dated December 1990, both of which are hereby incorporated by reference." Vasseur '879, 2:8-24.<br><br>"An intermediate network node often stores its routing information in a routing table maintained and managed by a routing information base (RIB). The routing table is a searchable data structure in which network addresses are mapped to their associated routing information. However, those skilled in the art will understand that the routing table need not be organized as a table, and alternatively may be another type of searchable data structure. Although the intermediate network node's routing table may be configured with a predetermined set of routing information, the node also may dynamically acquire ("learn") network routing information as it sends and receives data packets. When a packet is received at the intermediate network node, the packet's destination address (e.g., stored in a header of the packet) may be used to identify a routing table entry containing routing information associated with the received packet. Among other things, the packet's routing information indicates the packet's next-hop address." Vasseur '879, 2:25-41.<br><br>"Multi-Protocol Label Switching (MPLS) Traffic Engineering has been developed to meet data networking requirements such as guaranteed available bandwidth or fast restoration. MPLS Traffic Engineering exploits modem label switching techniques to build guaranteed bandwidth end-to-end tunnels through an IP/MPLS network of label switched routers (LSRs). These tunnels are a type of label switched path (LSP) and thus are generally referred to as MPLS Traffic Engineering (TE) LSPs. Examples of MPLS TE can be found in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels dated December 2001, RFC 3784 |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | entitled Intermediate-System-to-Intermediate-System (IS-IS) Extensions for Traffic Engineering (TE) dated June 2004, and RFC 3630, entitled Traffic Engineering (TE) Extensions to OSPF Version 2 dated September 2003, the contents of all of which are hereby incorporated by reference in their entirety." Vasseur '879, 2:58-3:6.<br><br>"Establishment of an MPLS TE-LSP from a head-end LSR to a tail-end LSR involves computation of a path through a network of LSRs. Optimally, the computed path is the "shortest" path, as measured in some metric, that satisfies all relevant LSP Traffic Engineering constraints such as e.g., required bandwidth, "affinities" (administrative constraints to avoid or include certain links), etc. Path computation can either be performed by the head-end LSR or by some other entity operating as a path computation element (PCE) not co-located on the head-end LSR. The head-end LSR (or a PCE) exploits its knowledge of network topology and resources available on each link to perform the path computation according to the LSP Traffic Engineering constraints. Various path computation methodologies are available including CSPF (constrained shortest path first). MPLS TE-LSPs can be configured within a single domain, e.g., area, level, or AS, or may also span multiple domains, e.g., areas, levels, or ASes." Vasseur '879, 3:7-24.<br><br>"The PCE is an entity having the capability to compute paths between any nodes of which the PCE is aware in an AS or area. PCEs are especially useful in that they are more cognizant of network traffic and path selection within their AS or area, and thus may be used for more optimal path computation. A head-end LSR may further operate as a path computation client (PCC) configured to send a path computation request to the PCE, and receive a response with the computed path, potentially taking into consideration other path computation requests from other PCCs. It is important to note that when one PCE sends a request to another PCE, it acts as a PCC." Vasseur '879, 3:25-36.<br><br>"Some applications may incorporate unidirectional data flows configured to transfer time-sensitive traffic from a source (sender) in a computer network to a destination (receiver) in the network in accordance with a certain "quality of service" (QoS). Here, network resources may be reserved for the unidirectional flow to ensure that the QoS associated with the data flow is maintained. The Resource ReSerVation Protocol (RSVP) is a network-control protocol that |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | enables applications to reserve resources in order to obtain special QoS for their data flows. RSVP works in conjunction with routing protocols to, e.g., reserve resources for a data flow in a computer network in order to establish a level of QoS required by the data flow. RSVP is defined in R. Braden, et al., Resource ReSerVation Protocol (RSVP), RFC 2205, the contents of which are hereby incorporated by reference in its entirety. In the case of traffic engineering applications, RSVP signaling (with Traffic Engineering extensions) is used to establish a TE-LSP and to convey various TE-LSP attributes to routers, such as border routers, along the TE-LSP obeying the set of required constraints whose path may have been computed by various means." Vasseur '879, 3:37-57.<br><br>"Generally, a tunnel is a logical structure that encapsulates a packet (a header and data) of one protocol inside a data field of another protocol packet with a new header. In this manner, the encapsulated data may be transmitted through networks that it would otherwise not be capable of traversing. More importantly, a tunnel creates a transparent virtual network link between two network nodes that is generally unaffected by physical network links or devices (i.e., the physical network links or devices merely forward the encapsulated packet based on the new header). While one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 3:58-4:3.<br><br>"Occasionally, a network element (e.g., a node or link) will fail, causing redirection of the traffic that originally traversed the failed network element to other network elements that bypass the failure. Generally, notice of this failure is relayed to the nodes in the network through an advertisement of the new network topology, e.g., an IGP or BGP Advertisement, and routing tables are updated to avoid the failure accordingly. Reconfiguring a network in response to a network element failure using, e.g., pure IP rerouting, can be time consuming. Many recovery techniques, however, are available to provide fast recovery and/or network configuration in the event of a network element failure, including, inter alia, "Fast Reroute", e.g., MPLS TE Fast Reroute. An example of MPLS TE Fast Reroute is described in Pan, et al., Fast Reroute Extensions to RSVP-TE for LSP Tunnels, RFC 4090, May 2005, which is hereby incorporated by reference as though fully set forth herein." Vasseur '879, 4:4-21. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "Fast Reroute (or FRR) has been widely deployed to protect against network element failures, where "backup tunnels" are created to bypass one or more protected network elements (e.g., links, shared risk link groups (SRLGs), and nodes). When the network element fails, traffic is quickly diverted ("Fast Rerouted") over a backup tunnel to bypass the failed element, or more particularly, in the case of MPLS, a set of primary TE-LSPs (tunnels) is quickly diverted. Specifically, the point of local repair (PLR) node configured to reroute the traffic inserts ("pushes") a new label for the backup tunnel, and the traffic is diverted accordingly. Once the failed element is bypassed, the backup tunnel label is removed ("popped"), and the traffic is routed along the original path according to the next label (e.g., that of the original TE-LSP). Notably, the backup tunnel, in addition to bypassing the failed element along a protected primary TE-LSP, also intersects the primary TE-LSP, i.e., it begins and ends at nodes along the protected primary TE-LSP." Vasseur '879, 4:22-39.<br><br>"To offer maximum protection, e.g., guaranteed bandwidth, during Fast Reroute, backup tunnels may reserve a configurable amount of bandwidth to ensure that Fast Rerouted traffic from the primary tunnel has a reserved path to follow. For example, the bandwidth reserved for the primary tunnel may also be reserved on the backup tunnel. While this approach provides maximum protection, it also requires a non-negligible amount of network resources (e.g., capacity/bandwidth) and may increase operational complexity." Vasseur '879, 4:40-48.<br><br>"Certain techniques are available to efficiently minimize the amount of resources required by the establishment and maintenance of the backup tunnels for Fast Reroute. One such technique is to create zero-bandwidth ("0-BW") backup tunnels (i.e., tunnels that reserve no bandwidth) to protect non-0-BW primary tunnels. This "best effort" approach does not guarantee that the path followed by the backup tunnel will have enough bandwidth to support the diverted primary tunnel at the time of failure without QoS degradation, however in many situations the path quality of the backup tunnel is sufficient. For instance, if the network is not overly congested, or the backup tunnel follows a non-congested path, there may be enough available bandwidth along the backup tunnel to support the newly rerouted traffic. Also, because primary tunnels often reserve bandwidth in response to "peak" traffic utilization, the amount of traffic over the primary tunnel at the time of failure may be far less than the reserved bandwidth (e.g., at "off-peak" times). Notably, those skilled in the art will understand that in |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | the absence of the above exceptions, a 0-BW backup tunnel may have unacceptable bandwidth (e.g., affecting path quality) to support the diverted traffic." Vasseur '879, 4:49-5:2.<br><br>"Currently, head-end nodes (LSRs) may be configured to systematically reroute the primary tunnels affected by the network element failure (e.g., diverted primary tunnels), especially in the case with 0-BW backup tunnels, such as, e.g., by reestablishing a new primary tunnel that follows a path excluding the failed network element. In particular, 0-BW backup tunnels represent a best effort attempt to allow the head-end node to more gracefully reestablish the primary tunnel in response to a failure, since the backup tunnels may not be able to support the diverted traffic without QoS degradation. The systematic reestablishing may potentially result in the reestablishment of a large number of primary tunnels (e.g., up to 3000 for a single network element failure in today's networks). Notably, reestablishing diverted primary tunnels may be undesirable for the network, such as by creating traffic churn, jitter, control plane overloads, etc., as will be understood by those skilled in the art. However, as noted above, there are situations where the backup tunnel may provide acceptable bandwidth, at least, for example, for a period of time (e.g., possibly short) until the failed network element is restored. In these situations, then, it may have been unnecessary to reestablish the diverted primary tunnels. There remains a need, therefore, for a technique that dynamically determines whether to reestablish a diverted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 5:3-28.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 5:32-47. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as Resource ReSerVation Protocol (RSVP) Traffic Engineering (TE) signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or type/length/value (TLV) encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object of the RSVP messages to convey the path quality notification." Vasseur '879, 5:48-58.<br><br>"In accordance with one aspect of the present invention, the head-end node requests the establishment of the primary tunnel (e.g., a TE-Label Switched Path, TE-LSP), along with a request for Fast Reroute protection of one or more network elements (e.g., with zero-bandwidth, 0-BW backup tunnels) at a PLR node. In addition, the head-end node may include a request for backup tunnel path quality notification, such as, e.g., through the use of the novel Feedback sub-object. The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and sends an error message (e.g., an RSVP PathErr) to the head end node indicating the "Fast Rerouting" of the primary tunnel." Vasseur '879, 5:59-6:6.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, the PLR node may collect metrics/statistics (e.g., packet drops, path cost, jitter, etc.) of the primary and/or backup tunnels. Once the primary tunnel is Fast Rerouted, the PLR node continues to collect metrics of the backup tunnel, and may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. Notably, the PLR node may be configured to send path quality notifications to the head-end node once, continually, periodically, in response to configurable changes in path quality, etc. Also, as in the case where multiple primary tunnels are Fast Rerouted, the path quality notification may include an indication of which Fast Rerouted primary tunnels in particular have been effected by the changed path quality." Vasseur '879, 6:7-23. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "In accordance with yet another aspect of the present invention, upon receiving the error message (PathErr), the head-end node may wait for the path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. Notably, in the event the head-end node does not receive a path quality notification for the backup tunnel (e.g., within a configurable time limit), the head-end node may attempt to reestablish the new primary tunnel. Moreover, where the head-end node has multiple primary tunnels being Fast Rerouted, a configurable subset of the primary tunnels may be reestablished, e.g., to reduce congestion of the backup tunnels, and/or to limit the number of reestablished primary tunnels within a given period of time." Vasseur '879, 6:24-43.<br><br>"Advantageously, the novel technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. By providing the head-end node of the primary tunnel with path quality feedback of the backup tunnel, the novel technique avoids reestablishing a potentially large number of tunnels over one or more alternate paths after a failure (and Fast Reroute) if the backup tunnels have acceptable path quality. In particular, the backup tunnels, e.g., 0-BW backup tunnels, may not be congested or subsequently burdened by the Fast Rerouted traffic of the primary tunnel. Also, the failed network element (thus the primary tunnel) may be quickly restored; therefore by not reestablishing the primary tunnel, network jitter, churn, etc., may be avoided. Further, the dynamic nature of the novel technique alleviates the need for cumbersome manual configuration." Vasseur '879, 6:44-59.<br><br>"FIG. 3 is schematic block diagram of an exemplary signaling (RSVP) message that may be advantageously used with the present invention." Vasseur '879, 7:6-8.<br><br>"FIG. 5 is a schematic block diagram of the computer network in FIG. 1 showing Fast Reroute protection of a primary tunnel using a backup tunnel in accordance with the present invention." Vasseur '879, 7:12-15. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 6 is a schematic block diagram of the computer network in FIG. 5 showing an unacceptable backup tunnel path quality and resultant reestablishing of the primary tunnel in accordance with the present invention." Vasseur '879, 7:16-19.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention." Vasseur '879, 7:20-24.<br><br>"FIG. 1 is a schematic block diagram of an exemplary computer network 100 comprising a plurality of nodes A-F, such as routers or other network devices, interconnected as shown. The nodes may be a part of one or more autonomous systems, routing domains, or other networks or subnetworks. For instance, routers A and E may be provider edge (PE) devices of a provider network, (e.g., a service provider network) that are interconnected to one or more customer networks through customer edge (CE) devices (not shown, while the remaining nodes B-D and F may be core provider (P) devices, as will be understood by those skilled in the art. Those skilled in the art will also understand that the nodes A-F may be any nodes within any arrangement of computer networks, and that the view shown herein is merely an example. For example, the nodes may be configured as connections to/from one or more virtual private networks (VPNs), as will be understood by those skilled in the art. These examples are merely representative. Those skilled in the art will understand that any number of routers, nodes, links, etc. may be used in the computer network 100 and connected in a variety of ways, and that the view shown herein is for simplicity." Vasseur '879, 7:29-49.<br><br>"Data packets may be exchanged among the computer network 100 using predefined network communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Asynchronous Transfer Mode (ATM) protocol, Frame Relay protocol, Internet Packet Exchange (IPX) protocol, etc. Routing information may be distributed among the routers of the computer network using predetermined Interior Gateway Protocols (IGPs), such as conventional distance-vector protocols or, illustratively, link-state protocols, through the use of IGP Advertisements." Vasseur '879, 7:50-60. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 2 is a schematic block diagram of an exemplary router 200 that may be advantageously used with the present invention, e.g., as an edge router or a core router. The router comprises a plurality of network interfaces 210, a processor 220, and a memory 240 interconnected by a system bus 250. The network interfaces 210 contain the mechanical, electrical and signaling circuitry for communicating data over physical links coupled to the network 100. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols, including, inter alia, TCP/IP, UDP, ATM, synchronous optical networks (SONET), wireless protocols, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), etc." Vasseur '879, 7:61-8:6.<br><br>"The memory 240 comprises a plurality of storage locations that are addressable by the processor 220 and the network interfaces 210 for storing software programs and data structures associated with the present invention. The processor 220 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures. A router operating system 242 (e.g., the Internetworking Operating System, or IOS™, of Cisco Systems, Inc.), portions of which is typically resident in memory 240 and executed by the processor, functionally organizes the router by, inter alia, invoking network operations in support of software processes and/or services executing on the router. These software processes and/or services may comprise routing services 247, Traffic Engineering (TE) services 244, and RSVP services 249. It will be apparent to those skilled in the art that other processor and memory means, including various computer-readable media, may be used to store and execute program instructions pertaining to the inventive technique described herein." Vasseur '879, 8:7-26.<br><br>"Routing services 247 contain computer executable instructions executed by processor 220 to perform functions provided by one or more routing protocols, such as IGP (e.g., OSPF and IS-IS), IP, BGP, etc. These functions may be configured to manage a forwarding information database (not shown) containing, e.g., data used to make forwarding decisions. Routing services 247 may also perform functions related to virtual routing protocols, such as maintaining VRF instances (not shown) as will be understood by those skilled in the art." Vasseur '879, 8:27-36. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "RSVP services 249 contain computer executable instructions for implementing RSVP and processing RSVP messages in accordance with the present invention. RSVP is described in RFC 2205, entitled Resource ReSerVation Protocol (RSVP), and in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels, both as incorporated above." Vasseur '879, 8:37-42.<br><br>"TE services 244 contain computer executable instructions for operating TE functions in accordance with the present invention. Examples of Traffic Engineering are described in RFC 3209, RFC 3784, and RFC 3630 as incorporated above, and in RFC 3473, entitled, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions dated January 2003, which is hereby incorporated by reference in its entirety. A TE database (TED, not shown) may be illustratively resident in memory 240 and used to store TE information provided by the routing protocols, such as IGP, BGP, and/or RSVP (with TE extensions, e.g., as described herein), including, inter alia, path quality information as described herein. The TED may be illustratively maintained and managed by TE services 244." Vasseur '879, 8:43-57.<br><br>"Changes in the network topology may be communicated among routers 200 using a link-state protocol, such as the conventional OSPF and IS-IS protocols. Suppose, for example, that a communication link fails or a cost value associated with a network node changes. Once the change in the network's state is detected by one of the routers, that router may flood an IGP Advertisement communicating the change to the other routers in the network. In this manner, each of the routers eventually "converges" to an identical view of the network topology." Vasseur '879, 8:58-67.<br><br>"In one embodiment, the routers described herein are IP routers that implement Multi-Protocol Label Switching (MPLS) and operate as label switched routers (LSRs). In one simple MPLS scenario, at an ingress to a network, a label is assigned to each incoming packet based on its forwarding equivalence class before forwarding the packet to a next-hop router. At each router, a forwarding selection and a new substitute label are determined by using the label found in the incoming packet as a reference to a label forwarding table that includes this information. At the network egress, a forwarding decision is made based on the incoming label |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | but optionally no label is included when the packet is sent on to the next hop. In some network configurations, one hop prior to the network egress, a penultimate hop popping (PHP) operation may be performed. Particularly, because the hop prior to the network egress (the penultimate hop) is attached to the network egress, the label is no longer needed to assure that the traffic follows a particular path to the network egress. As such, the PHP-enabled device "pops" the labels from the traffic before forwarding the traffic to the network egress, e.g., using conventional or native (IP) routing, thereby alleviating the task of removing the labels at the network egress." Vasseur '879, 9:1-23.<br><br>"The paths taken by packets that traverse the network in this manner are referred to as label switched paths (LSPs) or Traffic Engineering (TE)-LSPs. An example TE-LSP is shown as the thick line and arrow (T1) between a head-end node (router A) and a tailend node (router E) in FIG. 1. Establishment of a TE-LSP requires computation of a path, signaling along the path, and modification of forwarding tables along the path. MPLS TE establishes LSPs that have guaranteed bandwidth under certain conditions. Illustratively, the TE-LSPs may be signaled through the use of the RSVP protocol (with Traffic Engineering extensions), and in particular, RSVP TE signaling messages. Notably, when incorporating the use of PCEs (described below), the path computation request (and response) between PCC and PCE can be exchanged in accordance with a protocol specified in Vasseur, et al., Path Computation Element (PCE) Communication Protocol (PCEP)—Version 1—<draft-vasseur-pce-pcep-02.txt>, Internet Draft, September 2005, the contents of which are hereby incorporated by reference in its entirety. It should be understood that the use of RSVP or PCEP serves only as an example, and that other communication protocols may be used in accordance with the present invention." Vasseur '879, 9:24-45.<br><br>"In accordance with RSVP, to request a data flow (TE-LSP) between a sender and a receiver, the sender may send an RSVP path request (Path) message downstream to the receiver along a path (e.g., a unicast route) to identify the sender and indicate e.g., bandwidth needed to accommodate the data flow, along with other attributes of the TE-LSP. The Path message may contain various information about the data flow including, e.g., traffic characteristics of the data flow. Also in accordance with the RSVP, a receiver establishes the TE-LSP between the sender and receiver by responding to the sender's Path message with a reservation request |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | (Resv) message. The reservation request message travels upstream hop-by-hop along the flow from the receiver to the sender. The reservation request message contains information that is used by intermediate nodes along the flow to reserve resources for the data flow between the sender and the receiver, to confirm the attributes of the TE-LSP, and provide a TE-LSP label. If an intermediate node in the path between the sender and receiver acquires a Path message or Resv message for a new or established reservation (TE-LSP) and encounters an error (e.g., insufficient resources, failed network element, etc.), the intermediate node generates and forwards a path or reservation error (PathErr or ResvErr, hereinafter Error) message to the sender or receiver, respectively." Vasseur '879, 9:46-10:2.<br><br>"FIG. 3 is a schematic block diagram of portions of a signaling message 300 (e.g., RSVP message, such as Path, Resv or Error) that may be advantageously used with the present invention. Message 300 contains, inter alia, a common header 310 and one or more signaling protocol specific objects 320, such as an LSP-ATTRIBUTE object 330. The common header 310 may comprise a source address 312 and destination address 314, denoting the origination and requested termination of the message 300. Protocol specific objects 320 contain objects necessary for each type of message 300 (e.g., Path, Resv, Error, etc.). For instance, a Path message may have a sender template object, Tspec object, Previous-hop object, etc. The LSP-ATTRIBUTE object 330, for instance, may be used to signal attributes and/or information regarding an LSP (tunnel). To communicate this information, LSP-ATTRIBUTE object 330 (as well as specific objects 320) may include various type/length/value (TLV) encoding formats and/or flags, as will be understood by those skilled in the art. An example of an LSP-ATTRIBUTE object is further described in Farrel, et al., Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE <draft-ietf-mpls-rsvpte-attributes-05.txt>, Internet Draft, May 2005, which is hereby incorporated by reference as though fully set forth herein. A Resv message, on the other hand, may have specific objects 320 for a label object, session object, filter spec object, etc., in addition to the LSP-ATTRIBUTE object 330. Error messages 300 (e.g., PathErr or ResvErr), may also have specific objects 320, such as for defining the type of error, etc." Vasseur '879, 10:3-31. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "It should be noted that in accordance with RSVP signaling, the state of the TE-LSP is refreshed on a timed interval, e.g., every thirty seconds, in which RSVP Path and Resv messages are exchanged. This timed interval is configurable by a system administrator. Moreover, various methods understood by those skilled in the art may be utilized to protect against route record objects (RROs) contained in signaling messages for a TE-LSP in the event security/privacy is desired. Such RRO filtering prevents a head-end node of the TE-LSP from learning of the nodes along the TE-LSP, i.e., nodes within the provider network." Vasseur '879, 10:4-42.<br><br>"Although the illustrative embodiment described herein is directed to MPLS, it should also be noted that the present invention may advantageously apply to Generalized MPLS (GMPLS), which pertains not only to packet and cell-based networks, but also to Time Division Multiplexed (TDM) and optical networks. GMPLS is well known and described in RFC 3945, entitled Generalized Multi-Protocol Label Switching (GMPLS) Architecture, dated October 2004, and RFC 3946, entitled Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control, dated October 2004, the contents of both of which are hereby incorporated by reference in their entirety." Vasseur '879, 10:43-55.<br><br>"To obviate delays associated with updating routing tables when attempting to avoid a failed network element (i.e., during convergence), some networks have employed MPLS TE Fast Reroute (FRR). MPLS Fast Reroute is a technique that may be used to quickly divert ("Fast Reroute") traffic around failed network elements in a TE-LSP. MPLS Fast Reroute is further described, for example, by Fast Reroute Extensions to RSVP-TE for LSP Tunnels, as incorporated by reference above. According to the technique, one or more network elements (e.g. links or nodes) in a network are protected by backup tunnels following an alternate path. If a failure occurs on a protected link or node, TE-LSPs (and consequently the traffic that they carry) are locally diverted onto an appropriate alternate path (e.g., a "backup tunnel") by the node immediately upstream from the failure. The backup tunnel acts as a Fast Reroute path for the primary TE-LSP and obviates delays associated with other measures, such as tearing down the primary TE-LSP after having gracefully diverted the TE-LSPs affected by the failure, should an alternate path around the failed network element exist. In the event of a |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | failure of a protected element the head-end node of the backup tunnel (or a "point of local repair," PLR node) may quickly begin diverting traffic over the backup tunnel with minimal disruption to traffic flow. Those skilled in the art will understand that MPLS Fast Reroute is one example of link or node failure protection, and that other known correction mechanisms may be used in accordance with the present invention. As mentioned above, however, the head-end node of the Fast Rerouted primary tunnel may attempt to reestablish the primary tunnel in response to learning of the protected element failure, particularly in the case where the backup tunnel is a zero-bandwidth (0-BW) tunnel. The attempt to reestablish the primary tunnel has conventionally been a systematic response to Fast Rerouting (diverting) of the primary tunnel, regardless of the path quality of the backup tunnel." Vasseur '879, 10:56-11:23.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 11:24-39.<br><br>"In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as RSVP TE signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or TLV encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object 330 of the RSVP messages 300 to convey the path quality notification." Vasseur '879, 11:40-49. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 4 is a schematic block diagram illustrating the format of an extension object (TLV) 400, such as a Fast Reroute Feedback object, that may be advantageously used with the present invention. The extension object (or sub-object) 400 is illustratively embodied as a TLV contained in an LSP-ATTRIBUTE object 330 of an RSVP message 300 and is extended to carry backup (and/or primary) tunnel path quality information. To that end, the "Feedback" object 400 is organized to include a Type field 405 containing a predetermined type value signifying the specific content of the object 400. The Length field 410 is a variable length value. The TLV encoded format may also comprise one or more non-ordered sub-TLVs 450 carried within the TLV "payload" (e.g. Value field 415), each having a Type field 455, Length field 460, and Value field 465. The fields of the TLV 400 and sub-TLV(s) 450 are used in a variety of manners, including as described herein, according to the present invention." Vasseur '879, 11:59-12:8.<br><br>"In accordance with one aspect of the present invention, the head-end node (e.g., router A) requests the establishment of the primary tunnel (e.g., T1), such as a TE-LSP. Notably, the head-end node may be a head-end node for multiple primary tunnels, as will be understood by those skilled in the art. Along with the primary tunnel establishment, the head-end node may also request Fast Reroute protection of one or more network elements (e.g., all intermediate network elements) at a PLR node (e.g., router B as shown). Note that each intermediate node along the primary tunnel may act as a PLR node, and that router B is shown merely for simplicity. Illustratively, the Fast Reroute protection may be embodied as one or more zero-bandwidth (0-BW) backup tunnels at the PLR node (e.g., BT1). Those skilled in the art will also understand that the PLR node may protect more than one primary tunnel originating at more than one corresponding head-end node (not shown)." Vasseur '879, 12:9-25.<br><br>"The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. For example, various connectivity verification protocols, such as, e.g., Bidirectional Forwarding Detection (BFD), IGP "Hello" packets, BGP KEEPALIVE messages, etc., may be used to detect a failure of a network element, as will be understood by those skilled in the art. Furthermore, other lower-layer failure detection mechanisms (e.g. optical or SONET/SDH alarms) may be used to detect a network element failure. Once failure is detected, the PLR node diverts the |

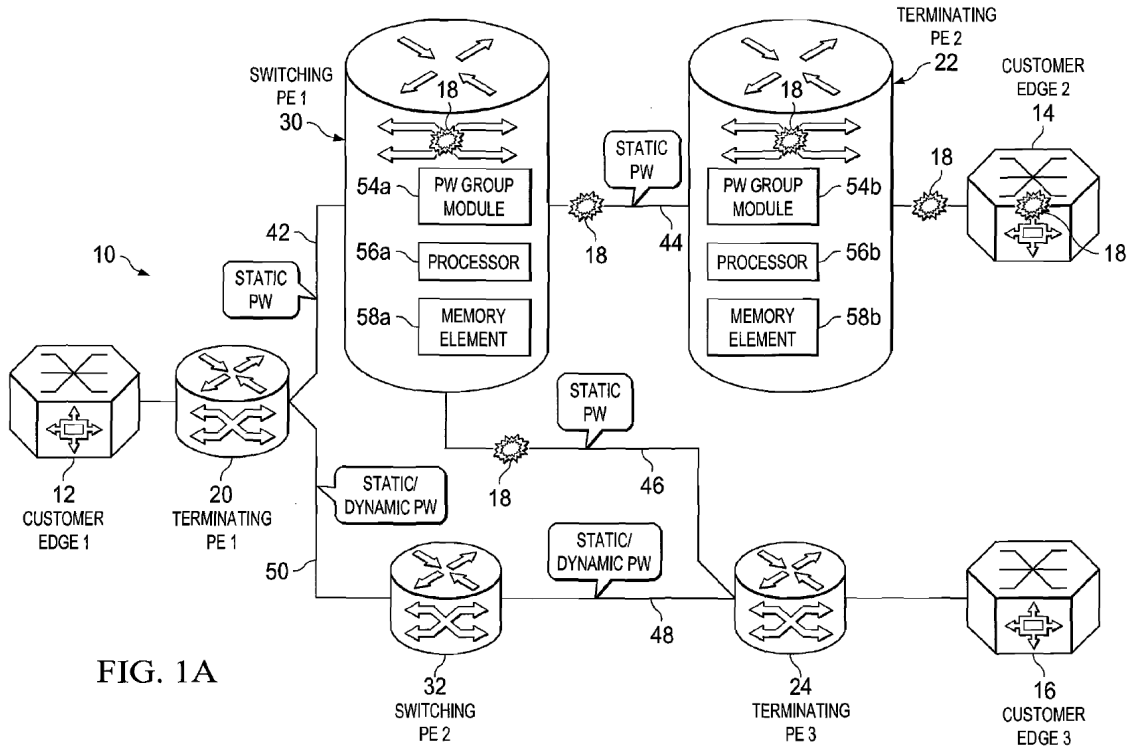
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | traffic onto the backup tunnel, and may send an error message (e.g., an RSVP PathErr 300, such as a conventional "tunnel locally repaired" message) to the head end node indicating the "Fast Rerouting" of the primary tunnel. FIG. 5 is a schematic block diagram of the computer network 100 in FIG. 1 showing Fast Reroute protection of the primary tunnel T1 (e.g., in response to a protected network element, router C, failure, indicated with an overlaid "X") using a backup tunnel BT1 in accordance with the present invention. Traffic originally received at the PLR node (router B) over the primary tunnel is now diverted over the backup tunnel to a remerge point (router D) of the primary tunnel, as will be understood by those skilled in the art." Vasseur '879, 12:42-65.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, each PLR node may collect metrics/statistics of the primary and/or backup tunnels. For instance, example metrics may comprise, inter alia, packet drops, path cost, jitter, delay, bandwidth, etc. The PLR node may collect the metrics through traffic monitoring, probes, independent calculations, and/or through cooperation with protected nodes of the primary tunnel (primary nodes) and nodes of the backup tunnel (backup nodes), e.g., transmitting path quality notifications. Once the primary tunnel is Fast Rerouted (i.e., diverted after failure of a protected network element), the PLR node continues to collect metrics of the backup tunnel. (Alternatively, metrics of the backup tunnel may be collected only after Fast Reroute, and not prior to Fast Reroute.)" Vasseur '879, 12:66-13:12.<br><br>"For example, a path cost increase from the primary tunnel to the backup tunnel may be computed (and transmitted, below) by the PLR node prior to the failure (or during the failure while the primary TE-LSP is diverted onto the backup tunnel) using its own routing tables. The path cost increase may be calculated as a difference between the entire length (head-end node to tail-end node) of the primary and backup tunnels, or just the difference between the protected segment of the primary tunnel and the backup tunnel (PLR node to remerge point). Also, a jitter increase of the primary and backup tunnels, which may be generally described as a difference between inter-arrival of packets, may be monitored using various known techniques, such as, e.g., sending probe packets (probes) from the PLR node to the remerge point. For instance, probes may determine that packets arrive at the remerge point along the primary tunnel from the PLR node consistently, e.g., every 10 milliseconds (ms) (e.g., an |

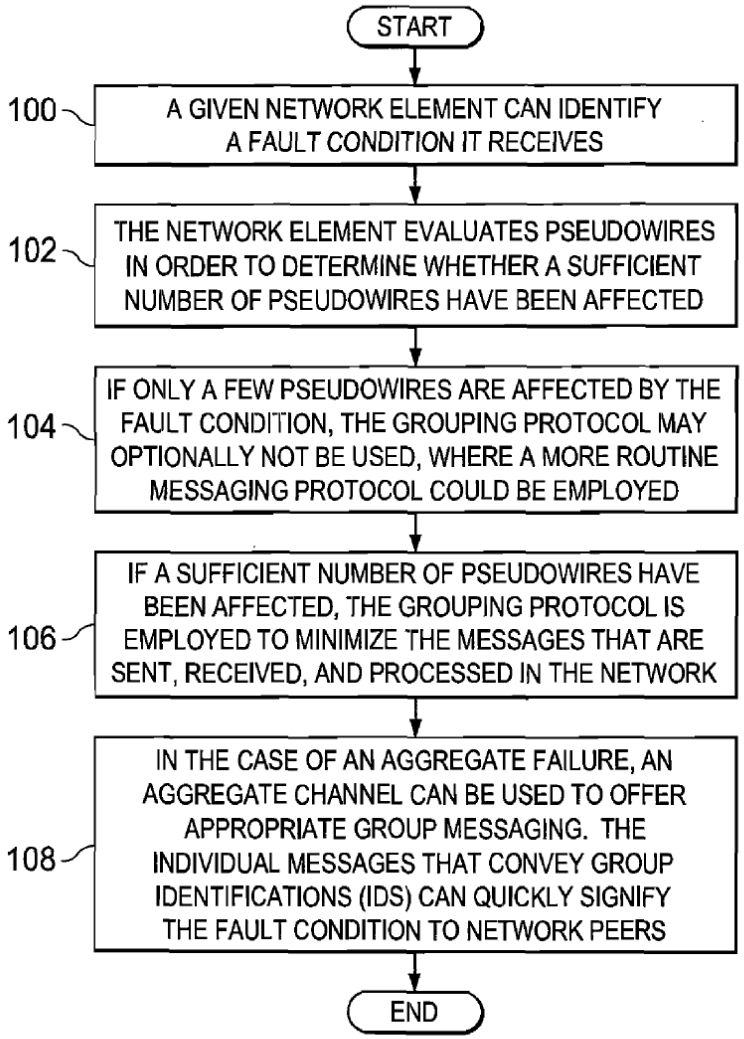
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | average value). After Fast Reroute, however, probes may determine that packets do not arrive at the remerge point along the backup tunnel from the PLR node consistently, e.g., one may arrive in 10 ms, another in 100 ms, another in 50 ms, etc. The non-constant rate of received packets (jitter) may be undesirable, e.g., in particular for voice over IP (VoIP) traffic, as will be understood by those skilled in the art." Vasseur '879, 13:13-36.<br><br>"As a further example, packet dropping may be monitored for the primary and backup tunnels prior to and after Fast Reroute (respectively). For instance, based on the tunnel label of the dropped packet, primary nodes and/or backup nodes may be able to distinguish which tunnel corresponds to the dropped packets. Each of the primary and/or backup nodes collect packet drop data, and periodically inform the PLR node of the number of dropped packets (e.g., though a corresponding Feedback object 400). In the case of a backup node, the PLR node receiving the notification may interpret the association of the backup tunnel label and the primary tunnel label to reference an appropriate primary tunnel. Those skilled in the art will understand that the above path quality metrics are merely examples, and that any other metrics/statistics useful for determining path quality of the backup tunnel may be used in accordance with the present invention (e.g., delay, bandwidth, etc.). Further, the path quality information may be measured and compared in a variety of manners, such as, e.g., as a difference between primary and backup tunnels before and after Fast Reroute, or simply the difference between the backup tunnel before and after Fast Reroute, etc." Vasseur '879, 13:37-58.<br><br>"Also after the primary tunnel is Fast Rerouted, the PLR node may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. For instance, the novel Feedback object 400 may include one or more sub-TLVs 450 corresponding to metrics/statistics, as described above. Notably, the path quality information pertaining to a particular metric/statistic may be transmitted as total values for interpretation by the head-end node (e.g., to determine the difference), or as PLR-node-computed differences (e.g., between the primary and backup tunnels, or before and after Fast Reroute). For instance, if the delay of the primary tunnel (along the protected segment) prior to Fast Reroute were 2 ms, and after Fast Reroute the delay of the backup tunnel were 5 ms, the notification may be configured to |

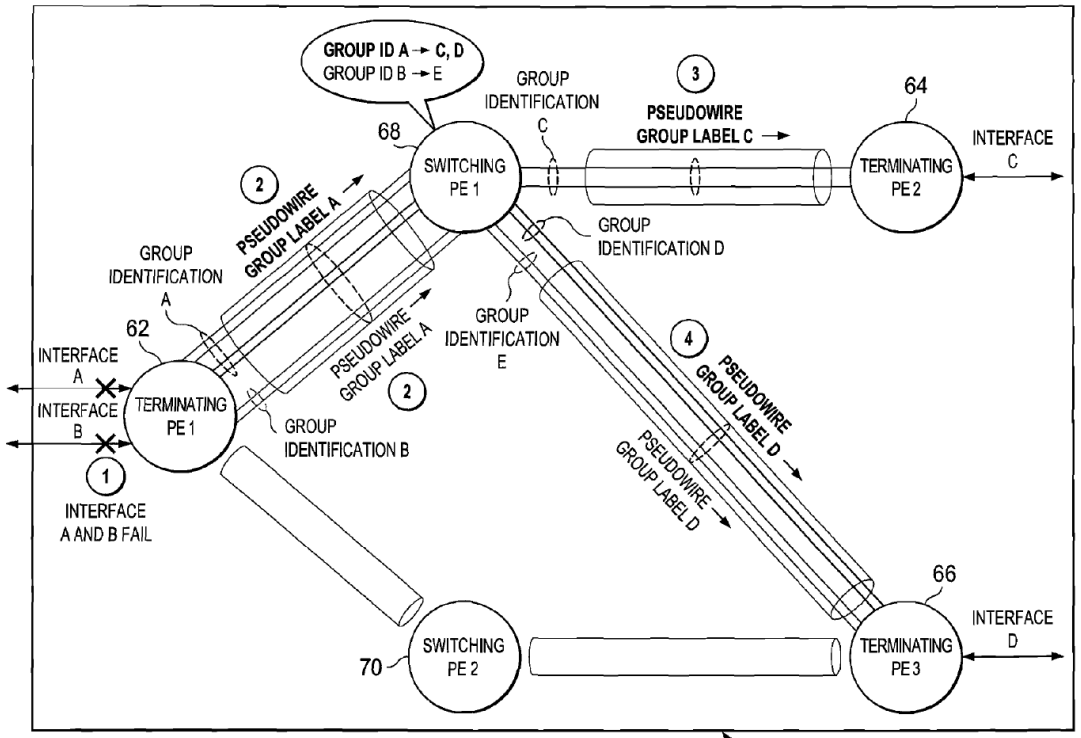
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | include both values 2 ms and 5 ms, or instead simply the difference, i.e., an increase of 3 ms." Vasseur '879, 13:59-14:8.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message 300 (PathErr), the head-end node may wait for at least one path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. Also, any number of metrics may be used in the determination, e.g., as configured by a system administrator. For example, using the metrics described above, a head-end node may be configured to reestablish the primary tunnel in response to i) a certain number of packet drops, ii) a percent increase in packet drops, iii) a number of packet drops and a percent increase in path cost, iv) a percent increase in path cost and a percent increase in jitter, etc. Those skilled in the art will understand that these are merely examples of possible path quality comparisons and reestablishment determinations, and that any comparisons to any metrics at any configurable changes may be used in accordance with the present invention." Vasseur '879, 14:41-59.<br><br>"If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. FIG. 6 is a schematic block diagram of the computer network 100 in FIG. 5 showing an unacceptable backup tunnel path quality (dotted line and arrow) and resultant reestablishment of the primary tunnel in accordance with the present invention. Those skilled in the art will understand that the reestablished primary tunnel may traverse one or more primary nodes (not shown), and that it may be computed specifically to avoid the failed network element and any network elements of the unacceptable backup tunnel. Those skilled in the art will also understand that the attempt to reestablish the primary tunnel may not be able to find an acceptable path, in which case the head-end node may continue to use the unacceptable backup tunnel or other unacceptable rerouted path." Vasseur '879, 14:60-15:9. |

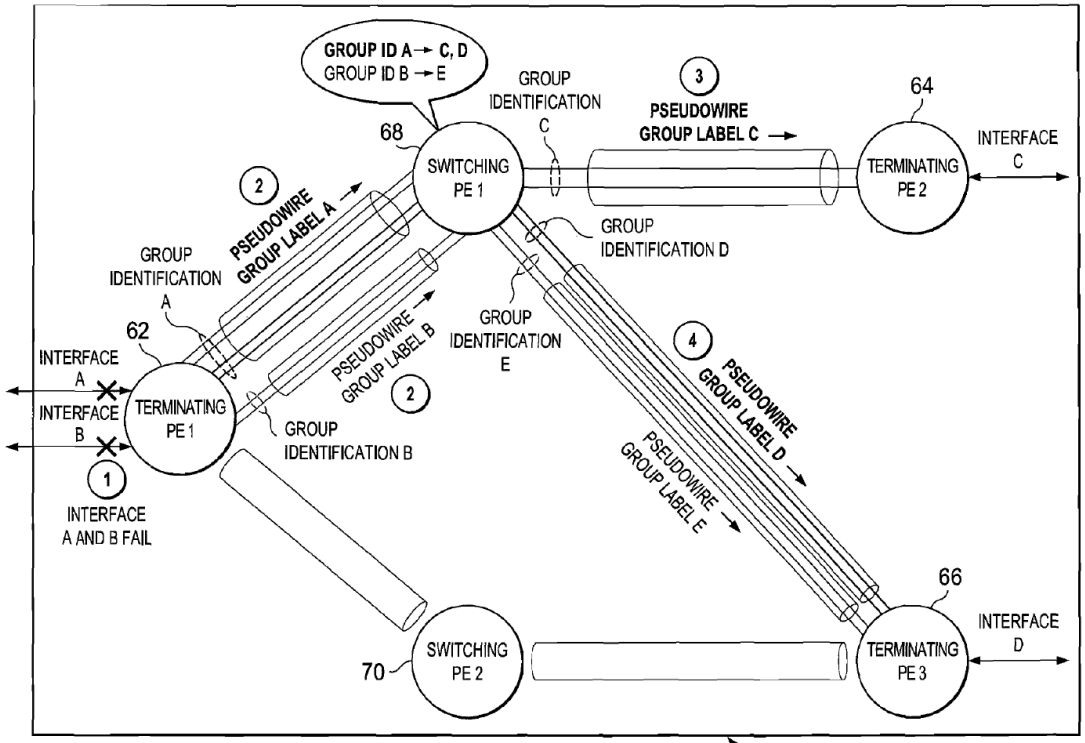
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention. The procedure 700 starts at step 705, and continues to step 710, where a head-end node (e.g., router A) establishes a primary tunnel (e.g., T1) to a destination tail-end node (e.g., router E). During or after establishment, the head-end node requests protection of the primary tunnel at step 715, and in response, PLR nodes along the primary tunnel (e.g., router B) create backup tunnels (e.g., BT1) to protect the primary tunnel in step 720. (Those skilled in the art will understand that backup tunnels around the protected network elements may already exist at the PLR node, and that "creating backup tunnels" in step 720 implies an association with pre-existing backup tunnels.) As mentioned above, these backup tunnels may illustratively be embodied as 0-BW backup tunnels. In accordance with the present invention, the head-end node may additionally request backup tunnel path quality notification from the PLR nodes in step 725, such as, e.g., through the use of empty corresponding Feedback objects in RSVP (Path) messages 300, as described above." Vasseur '879, 15:37-58.<br><br>"The procedure 700 continues to FIG. 7B (step "A"), where in step 750 the PLR node detecting the failure diverts ("Fast Reroutes") the primary tunnel traffic to the backup tunnel and sends an error message (e.g., an RSVP (Error) message 300) to the head end node, e.g., a "tunnel locally repaired" message. The detecting PLR node continues to collect path quality information for the backup tunnel in step 755 and at step 760 determines whether to send the path quality notification to the head-end node. For example, as mentioned above, the PLR node may be configured to continually send notifications, or periodically, or in response to a configurable change in path quality, etc. Also as mentioned above, the PLR node may be configured to send either the actual path quality information or the change (difference) in path quality. (As further mentioned above, metrics of the backup tunnel may alternatively be collected only after Fast Reroute, and not prior to Fast Reroute.)." Vasseur '879, 16:4-20.<br><br>"If the PLR node decides to send the notification in step 760, then the head-end node determines whether to reestablish the primary tunnel based on the backup tunnel path quality notification in step 765, e.g., based on one or more configurable thresholds, percentages, etc., as described above. If the backup tunnel is currently maintaining an acceptable quality for the |

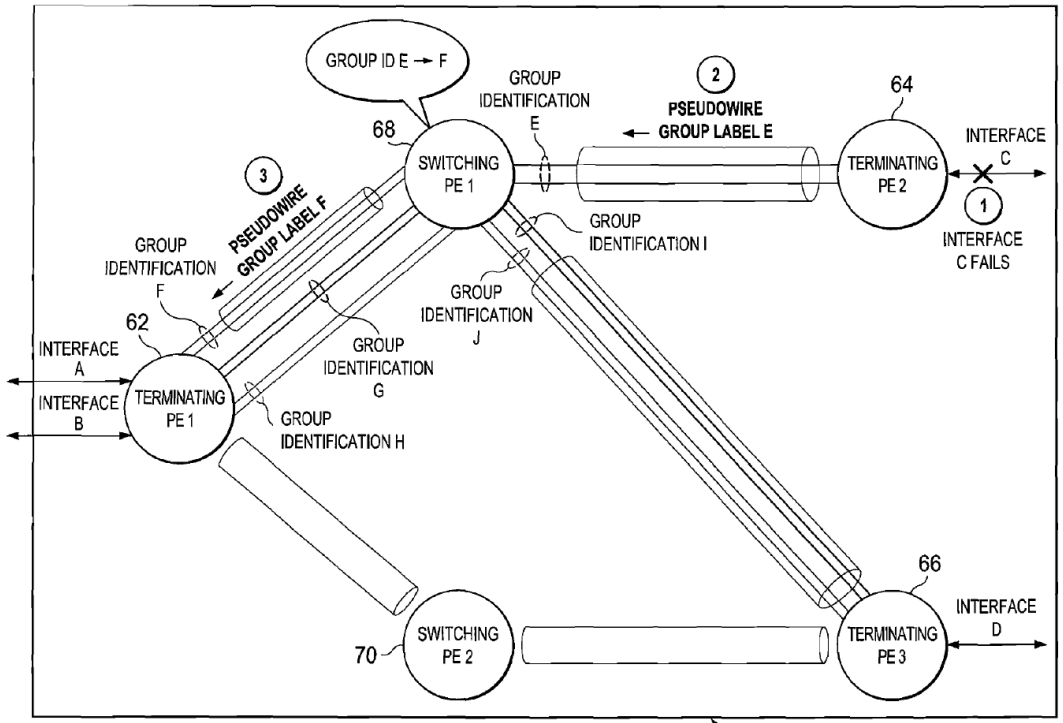
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | traffic flow in step 770, the head-end node may continue to utilize the backup tunnel, and the PLR node continues to collect path quality information in step 755 to detect any change in quality. Otherwise, if the backup tunnel quality is not acceptable in step 770, the head-end node may attempt to reestablish the primary tunnel in step 775. Notably, as mentioned above, if the PLR node has not sent any notification (step 760) within a configurable period of time in step 780, e.g., due to a backup tunnel failure, over-congestion, etc., then the head-end node may also attempt to reestablish the primary tunnel in step 775 accordingly. Moreover, as described above, in the event more than one primary tunnel is Fast Rerouted for the head-end node, various techniques to reestablish one or more of the primary tunnels may be used (e.g., as many tunnels as necessary, a configurable subset of tunnels, all tunnels, the congested tunnels, etc.). The procedure 700 ends in step 785." Vasseur '879, 16:21-43.<br><br>"While there has been shown and described an illustrative embodiment that dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the present invention. For example, the invention has been shown and described herein using "Fast Reroute," e.g., MPLS TE Fast Reroute (FRR). However, the invention in its broader sense is not so limited, and may, in fact, be used with other network element protection and failure correction mechanisms as will be understood by those skilled in the art. Moreover, while the above description describes performing the technique at the head-end node and PLR node, the invention may also be advantageously used with PCEs. In addition, while one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 16:63-17:13.<br><br>**Rustogi discloses**:<br>"An example method includes identifying a fault condition in a network, and evaluating pseudowires affected by the fault condition in order to make a determination as to whether an aggregate failure occurred in the network for a group of pseudowires. The method also includes communicating a group message indicating that the group of pseudowires is associated with the fault condition. The group message includes a group identification (ID), |

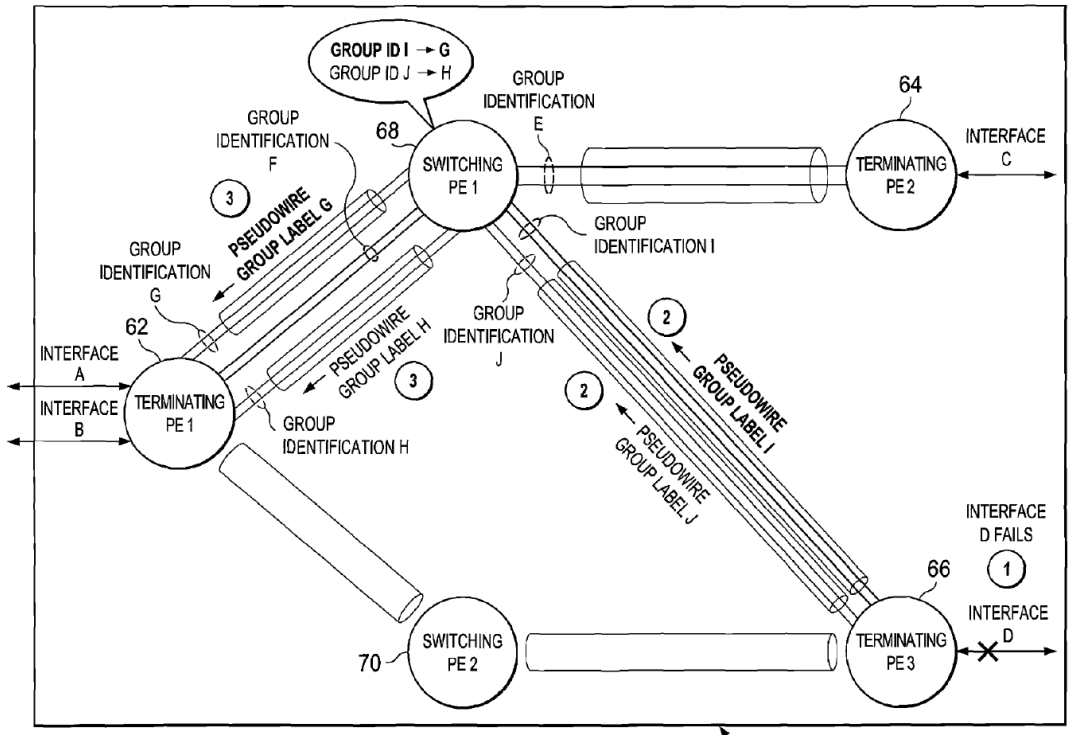
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | which identifies the group of pseudowires, and the group message includes a pseudowire group label identifying an in-band aggregate channel. More specifically, the pseudowire group label can be applicable to static pseudowires. In more detailed embodiments, the group ID identifies the group of pseudowires that are associated with an attachment circuit, a label switched path, or a port. Internal mappings can be maintained such that a plurality of pseudowires is mapped to individual interfaces of network elements in the network." Rustogi, Abstract.<br>  <br><br>FIG. 1A<br>Rustogi, FIG. 1A. |

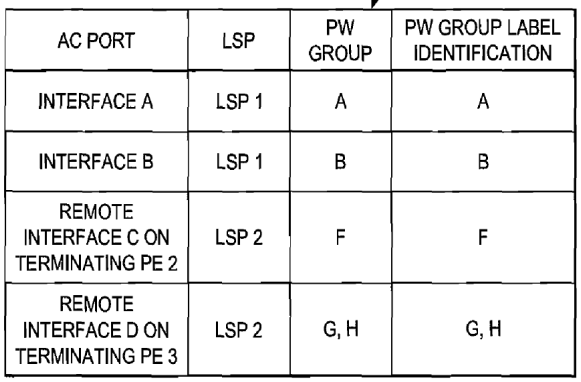
| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 1B<br>Rustogi, FIG. 1B. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Rustogi, FIG. 2. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Rustogi, FIG. 3. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Rustogi, FIG. 4. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Rustogi, FIG. 5. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | 74<br>  <br>FIG. 6<br>Rustogi, FIG. 6.<br><br>"The field of communications has become increasingly important in today's society. In particular, the ability to quickly and to effectively provision connections presents a significant challenge to component manufacturers, system designers, and network operators. Multiprotocol Label Switching (MPLS) is a mechanism in telecommunications networks that carries data from one network node to the next. Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be emulated over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDUs) and transmitting them over pseudowires. Protocols exist for establishing and maintaining the pseudowires. Certain issues have arisen in pseudowire scenarios, where faults are detected in the network." Rustogi, ¶ [0002].<br><br>"FIG. 1A is a simplified block diagram of a communication system for providing pseudowire group labels in a network environment in accordance with one embodiment of the present disclosure." Rustogi, ¶ [0004]. |

| AC PORT | LSP | PW GROUP | PW GROUP LABEL IDENTIFICATION |
|---|---|---|---|
| INTERFACE A | LSP 1 | A | A |
| INTERFACE B | LSP 1 | B | B |
| REMOTE INTERFACE C ON TERMINATING PE 2 | LSP 2 | F | F |
| REMOTE INTERFACE D ON TERMINATING PE 3 | LSP 2 | G, H | G, H |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "FIG. 1B is a simplified flowchart depicting one possible, generic operational flow associated with the communication system." Rustogi, ¶ [0005].<br><br>"FIG. 2 is a simplified block diagram of an example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0006].<br><br>"FIG. 3 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0007].<br><br>"FIG. 4 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0008].<br><br>"FIG. 5 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0009].<br><br>"FIG. 6 is a simplified table of an example set of pseudowire group provisioning parameters in accordance with one embodiment." Rustogi, ¶ [0010].<br><br>"FIG. 1A is a simplified block diagram of a communication system 10 for providing pseudowire group labels in accordance with one example of the present disclosure. FIG. 1A includes a customer edge 1 (CE1) 12, a CE2 14, and a CE3 16, where a number of faults 18 are shown as propagating in the network. Typically, when an error or a failure occurs in the network (e.g., an interface failure, a pulled cable, a switch failure, hardware/software failures generally, etc.), messages are sent to various network devices in order to inform them of these fault conditions. Faults 18 of FIG. 1A are indicative of such messages, where the underlying fault condition (being signaled by the messages) can occur virtually anywhere in a network (e.g., in a customer edge, in provider equipment, etc.). FIG. 1A also includes a terminating provider equipment 1 (TPE1) 20, a TPE2 22, a TPE3 24, a switching provider edge 1 (SPE1) 30, and a SPE2 32. In one particular example implementation, the TPEs and SPEs of FIG. 1A are switches that are configured to exchange data in a network environment." Rustogi, ¶ [0012]. |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | "SPE1 30 may include a pseudowire (PW) group module 54 a, a processor 56 a, and a memory element 58 a. In a similar fashion, TPE2 22 may include a pseudowire group module 54 b, a processor 56 b, and a memory element 58 b. FIG. 1A also includes a number of static pseudowires 42, 44, and 46. A set of static/dynamic pseudowires 48, 50 is also provided. Note that the group labeling protocol discussed herein can be executed between individual SPEs, TPEs, or between any combinations of these elements." Rustogi, ¶ [0013].<br><br>"In one particular arrangement, communication system 10 is provided in conjunction with a Layer-2 virtual private networks (L2VPN)/operation, administration, and maintenance (OAM) L2VPN/OAM framework. The OAM framework is intended to provide OAM layering across L2VPN services, pseudowires, and packet switched network (PSN) tunnels. Communication system 10 may include any suitable networking protocol or arrangement that provides a communicative platform for communication system 10. Thus, communication system 10 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Communication system 10 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs." Rustogi, ¶ [0014].<br><br>"Failure detection and failure notification for static pseudowires is inadequate, where sluggish signaling can result in poor scalability for failover performance. Typically, static pseudowires are manually configured at respective endpoints, where control channels are absent for providing group level signaling messages. Aggregate channels are significant tools for providing suitable scalability in the network, but no such aggregate channel exists for static pseudowires. For dynamic pseudowires, such an aggregate channel may be present in the form of a label distribution protocol (LDP) directed session. However, no such protocol exists for static pseudowire configurations such that an in-band aggregate channel would be available for static pseudowires." Rustogi, ¶ [0016].<br><br>"Communication system 10 can address the aforementioned issues (and others) by offering a pseudowire group label that can represent an aggregate channel for groups of static pseudowires. The aggregate channel of communication system 10 can allow for improved |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | scalability of failover performance. In accordance with one potential configuration of communication system 10, a pseudowire group label is representative of a group of static pseudowires transported over a label switched path (LSP). The pseudowire group label can identify the aggregate channel, which captures the hierarchy relevant to OAM mechanisms. Additionally, the groups represented by the group identification (ID) can be mutually exclusive, where a pseudowire is part of only one group. In other embodiments, a pseudowire can be part of multiple groups, or be configured in any other suitable manner based on particular network arrangements." Rustogi, ¶ [0017].<br><br>"During operations, and with brief reference to FIG. 1B, a given network element can identify a fault condition it receives (at step 100) and, subsequently, evaluate pseudowires in order to determine whether a sufficient number of pseudowires have been affected. This is reflected by step 102. If only a few pseudowires are affected by the fault condition, the grouping protocol outlined herein may have only nominal value, where there could be an option to simply communicate the fault condition in a more routine manner, as outlined in step 104. However, if a sufficient number of pseudowires have been affected, the grouping protocol outlined herein can be employed to minimize the messages that are sent, received, and processed in the network. This is reflected as step 106. Note that the determination (as to whether a sufficient number of pseudowires have been impacted by the fault condition) can involve accessing internal tables such that a quick mapping can occur to determine if an aggregate failure has occurred. As used herein, the term 'aggregate failure' simply connotes that a sufficient number of pseudowires have experienced the fault condition such that an aggregate channel can be employed to offer appropriate group messaging. For the aggregate failure condition, the individual messages that convey Group identifications (IDs) can quickly signify the fault condition to network peers, as shown in step 108." Rustogi, ¶ [0018].<br><br>"In specific regards to OAM mechanisms, OAM messages typically result from common failures in the network. These fault conditions can be aggregated such that they are signaled as a single message, which could represent a group of failed pseudowires (as opposed to sending individual messages for each failed pseudowire). Hence, a single message could be sent to represent all the relevant OAM messages propagating in communication system 10. The group label that propagates in communication system 10 provides an architecture with a |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | significant level of aggregation for failed pseudowires (i.e., pseudowires being affected by a given fault condition), particularly for OAM messaging. Moreover, the in-band aggregate channel of communication system 10 is based (at least in part) on the evolving trends of OAM mechanisms, which are required to be fast, responsive, and capable of being implemented in hardware or software. Additionally, in-band OAM protocols are a better measure of the path availability." Rustogi, ¶ [0020].<br><br>"In operation of one example implementation, a group label can represent the tuple <attachment circuit (AC) port level grouping, LSP>. This could signify that all pseudowires on an AC port (sought for aggregation) traverse a given LSP. Multiple pseudowire groups can exist within an LSP. Similarly, pseudowires on the same AC port (that traverse a different LSP) can use a different pseudowire group label. Alternatively, an administrator may seek to employ a one-to-one mapping between an LSP and a group label. If that were the case, then only one pseudowire group would exist within an LSP. In scenarios where there is no LSP label in the packet (e.g., due to penultimate hop popping), the pseudowire group label can provide the hierarchy that is appropriate." Rustogi, ¶ [0021].<br><br>"In one particular example, the group level pseudowire OAM message can be sent with the following label stack: Explicit/Implicit LSP Label+pseudowire group Label+GAL+ACH+pseudowire OAM with grouping TLV (where GAL=Generic Associated Channel Label, ACH=Associated Channel Header, TLV=Type-Length-Value). If there are multiple LSPs, then one group label can be provisioned for each LSP (for each pseudowire group), where per group messages can be sent on each LSP. The group label does not necessarily have a one-to-one mapping to the grouping of pseudowires implied by the Group ID in the grouping TLV. Note also that the group-based aggregate channel is applicable to static pseudowires, as well as for dynamic pseudowires in certain applications." Rustogi, ¶ [0022].<br><br>"As discussed herein, the aggregate channel of communication system 10 can be configured in various ways. For example, and with regards to a first option, a separate label may simply be used to identify a pseudowire group within an LSP. The association of an OAM message and a pseudowire group is straightforward. There could potentially be multiple pseudowire |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | group labels per LSP. As a second option, one group label can be used to identify a common pseudowire group channel on the LSP. In this implementation, one pseudowire group label is provided per LSP. The OAM message association to a pseudowire group is not as simple as the first option. As a third option, one pseudowire is simply designated to convey grouping information (e.g., without using a group label). In this case, there is no need for a pseudowire group label. Again, the OAM message association to a pseudowire group is not as simple as the first option." Rustogi, ¶ [0023].<br><br>"Any combination of formatting (for the Group ID and the pseudowire group label) can be used in the group message to be communicated in the network. In one example, only one of these elements is communicated when an aggregate fault condition is detected, or these elements can be combined into a single unique identifier. In the most generic example, a group message would at least include the Group ID (identifying the pseudowires affected by the fault) and a pseudowire group label (identifying an aggregate channel for communicating the group message). In this generic sense, a pipe (the Group ID) within a pipe (the pseudowire group label) is being identified, where the group message is identifying both elements during an aggregate fault condition. Operational details of communication system 10 are described below with reference to FIGS. 2-6. Note that before turning to additional example flows and example embodiments of the present disclosure, a brief overview of the infrastructure of communication system 10 is provided." Rustogi, ¶ [0024].<br><br>"CE1 12, CE2 14, and CE3 16 represent devices, infrastructure, equipment, clients, or customers seeking to initiate a data session in communication system 10. These elements may can comprise a digital subscriber line access multiplexer (DSLAM), a router, a personal computer, a server, a switch, and/or other devices associated with data propagation. Further, these elements may sit behind, or in front of, one or more of these identified devices. The term 'CE' may be inclusive of the devices identified above (e.g., a DSLAM, a switch, etc.), as well as devices used to initiate a communication, such as a console, a proprietary endpoint, a telephone, a cellular telephone, a bridge, a computer, a personal digital assistant (PDA), a laptop or an electronic notebook, or any other device, component, element, or object capable of initiating voice, audio, media, or data exchanges within communication system 10. The customer element may also include any device that seeks to initiate a communication on behalf |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating a voice, a video, text, or a data exchange within communication system 10. Data, as used herein in this document, refers to any type of video, numeric, voice, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another." Rustogi, ¶ [0025].<br><br>"SPE1 30, SPE2 32, TPE1 20, TPE2 22, and TPE3 24 are network elements that facilitate communications in two directions in a network environment. In one particular example, each of these network elements is a switch configured to exchange data over static and/or dynamic pseudowire links. Further, the traffic exchanged between these components may be directed over an MPLS transport in certain embodiments. As used herein in this Specification, the term 'network element' is meant to encompass switches, routers, bridges, gateways, servers, processors, loadbalancers, firewalls, or any other suitable device, component, element, or object operable to exchange or process information in a network environment. Moreover, these network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information. Along similar design alternatives, any of the internal modules and components of these network elements may be combined in various possible configurations." Rustogi, ¶ [0029].<br><br>"Turning to FIG. 2, FIG. 2 is a simplified block diagram of an example system 60 for providing an example use case using per-label switched path (LSP) pseudowire group labels. FIG. 2 includes a TPE1 62, a TPE2 64, a TPE3 66, a SPE1 68, and a SPE2 70. Each pseudowire group is identified, where a group identification (ID) for Group A and Group B is depicted at TPE1 62. Similarly, Groups C, D, and E have Group IDs at SPE1 68. TPE2 64 and TPE3 66 can couple to interfaces C and D, respectively." Rustogi, ¶ [0031].<br><br>"In this particular example, interfaces A and B have failed. Note that there is a multitude of attachment circuits (e.g., 1000 attachment circuits) that are being transported over these interfaces A and B, where the attachment circuits are being tunneled into a corresponding |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | number of pseudowires. For example, there could be 500 attachment circuits on interface A (implicating 500 pseudowires) and 500 attachment circuits on interface B, where the fault condition for the pseudowires should be signaled. In other flawed systems, an architecture would individually signal this fault condition for each pseudowire (e.g., via signaling between TPE1 62 and SPE1 68). Instead of sending 500 messages, a single message can be sent, where a single label (and Group ID) can be used to identify the pseudowires. In this case, the Group ID A is used to signal the fault condition for 300 pseudowires and for 200 pseudowires (i.e., the top two links connecting TPE1 62 and SPE1 68) using a single message (that includes Group Label A and Group ID A). Thus, the status for Group A is quickly communicated to SPE1 68. Similarly, Group ID B can be used to signal the status of the other 500 pseudowires to appropriately convey the status for Group B. More specifically, the message can include Group Label A and Group ID B. Note that all 1000 pseudowires have effectively been accounted for using these Group IDs A and B." Rustogi, ¶ [0032].<br><br>"FIG. 3 is a simplified block diagram of an example system 72 for providing another use case for pseudowire group labels. Note that the grouping mechanism outlined herein is not limited to pseudowires that propagate over LSPs. Certain pseudowires can propagate over an LSP and represent one group, where two ports can be provisioned for two different groups (e.g., Group A and Group B). Hence, FIG. 3 is depicting a use case using pseudowire group labels for <port, LSP>mapping. In a general sense, such a configuration is showing how pseudowire mechanics can be used to offer different group signaling, which may be based on various possible implementations. Thus, there is a group level construct corresponding to the group labels that are created such that any OAM protocol can send the appropriate aggregate messages. In this particular example, the signaling for Group ID A, B, C, and D is similar to that of FIG. 2; however, the grouping mechanism has simply changed." Rustogi, ¶ [0035].<br><br>"FIG. 4 is a simplified block diagram of an example system 76 for providing another use case for pseudowire group labels. In this particular example, interface C fails (as shown at TPE2 64). Note that the same logical flow occurs in FIG. 4 in terms of the group signaling, as previously discussed. The group labels in two directions do not have to be the same, where the groupings for the messaging are not necessarily symmetrical. In this particular example, TPE2 64 sends a status for Group E with the corresponding group label (i.e., Group ID E for |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | 300 pseudowires), where that message will have a Group Label E and a Group ID E. Hence, this particular signaling is indicative of 300 pseudowires failing in the network. SPE1 68 can send the status for Group F (where the Group ID F is associated with 300 pseudowires) to TPE1 62, where that message includes a Group Label F and a Group ID F." Rustogi, ¶ [0036].<br><br>"FIG. 5 is a simplified block diagram of an example system 80 for providing another use case for pseudowire group labels. In this particular example, interface D fails (as shown at TPE3 66), where all 700 pseudowires fail. In one implementation, TPE3 66 does not have a 700 pseudowire Group ID. Instead, the Group IDs can correspond to 200 and 500 pseudowires, when summed together account for the 700 pseudowires. In this particular example, TPE3 66 sends one message for Group I (representing 200 pseudowires) and another message for Group J (representing 500 pseudowires) to SPE1 68. In response, SPE1 68 sends a message for Group G (representing 200 pseudowires) and another message for Group H (representing 500 pseudowires). Again, the signaling being exchanged between these elements is minimal due to the effective grouping of pseudowires. SPE1 68 also sends a single message for Group I (associated with 200 pseudowires) and Group J (associated with 500 pseudowires) to TPE3 66, which is coupled to interface D. Group ID G is associated with 200 pseudowires, whereas Group ID H is representative of 500 pseudowires." Rustogi, ¶ [0037].<br><br>"FIG. 6 is a simplified table 74 illustrating an example set of pseudowire group provisioning parameters for TPE1 62, where these particular provisioning parameters could be relevant to the configuration of FIG. 3. At least in one generic sense, FIG. 2 can reflect one approach for mapping a PW group label to a PW Group ID, while FIGS. 3-5 can reflect a second approach for such mappings, where table 74 is associated with that second approach." Rustogi, ¶ [0038].<br><br>"In particular, table 74 illustrates the mapping between SPE1 68 and TPE1 62. The first column represents the attachment circuit port (e.g., interface A, interface B, remote interface C on TPE2 64, and remote interface D on TPE3 66). Additionally, table 74 depicts a number of LSPs, a set of pseudowire grouping labels, and a set of pseudowire Group IDs. Note that the Group IDs are provided inside the pseudowire group labels in this example such that these two columns match in table 74. Additionally, note that table 74 is merely representing some |

| No. | '821 Patent Claim 7 | The Reference |
|---|---|---|
| | | of the possible characteristics in a single direction, where different constructs could be used in the reverse direction. Note that the provisioning as discussed herein can significantly reduce messaging such that these presented concepts offer increased scalability. This is due in part to the nominal processing that occurs in the network, in contrast to the processing required to evaluate a prolific amount of signaling messages associated with particular pseudowires. Additionally, the paradigm discussed herein can afford service providers an adequate amount of downtime after a failure has occurred in the network." Rustogi, ¶ [0039]. |

| No. | '821 Patent Claim 8 | The Reference |
|---|---|---|
| 8 | The method of claim 4, further comprising the step of configuring said working entity as revertive. | The Reference discloses the method of claim 4, further comprising the step of configuring said working entity as revertive.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 9 | The Reference |
|---|---|---|
| 9[preamble] | The method of claim 4, wherein said overall cost function comprises: | The Reference discloses the method of claim 4, wherein said overall cost function comprises.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 4. |

| No. | '821 Patent Claim 9 | The Reference |
|---|---|---|
| 9[a] | selecting a subset of entity pairs wherein each entity pair of said subset has substantially minimum probability of a concurrent failure of said protection entity and said working entity; and | The Reference discloses selecting a subset of entity pairs wherein each entity pair of said subset has substantially minimum probability of a concurrent failure of said protection entity and said working entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |
| 9[b] | if said subset comprises at least two entity pairs, selecting an entity pair from said subset that minimizes an entity cost function. | The Reference discloses if said subset comprises at least two entity pairs, selecting an entity pair from said subset that minimizes an entity cost function.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references.<br><br>**Kurose discloses**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path.<br><br>"The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a |

| No. | '821 Patent Claim 9 | The Reference |
|---|---|---|
| | | packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

| No. | '821 Patent Claim 9 | The Reference |
|---|---|---|

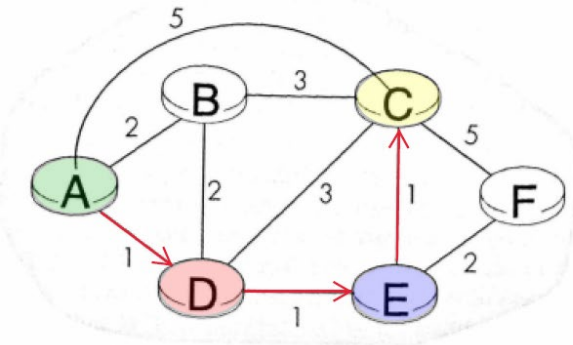



**Figure 4.4** ✦ Abstract model of a network

- the first link in the path is connected to the source
- the last link in the path is connected to the destination
- for all $i$, the $i$ and $i-1$st link in the path are connected to the same node
- for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).

In Figure 4.4, for example, the least-cost path between nodes *A* (source) and *C* (destination) is along the path *ADEC*. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)

As a simple exercise, try finding the least-cost path from nodes *A* to *F*, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from *A* to *F* by examining Figure 4.4, tracing a few routes from *A* to *F*, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between *A* and *F*? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:

Kurose at 281.

| No. | '821 Patent Claim 9 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |

| No. | '821 Patent Claim 10 | The Reference |
|---|---|---|
| 10 | The method of claim 9, wherein said entity cost function comprises a predefined metric. | The Reference discloses the method of claim 9, wherein said entity cost function comprises a predefined metric.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references.<br><br>**Kurose discloses**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path.<br><br>"The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current |

| No. | '821 Patent Claim 10 | The Reference |
|---|---|---|
| | | purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

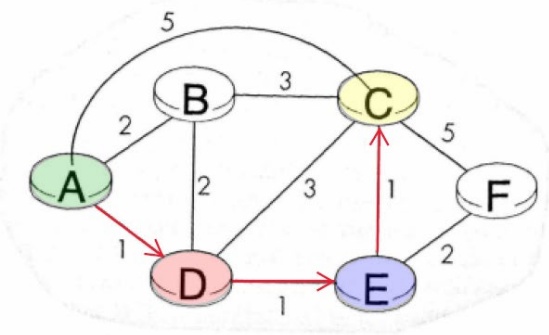



**Figure 4.4** ✦ Abstract model of a network

- the first link in the path is connected to the source
- the last link in the path is connected to the destination
- for all $i$, the $i$ and $i-1$st link in the path are connected to the same node
- for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).

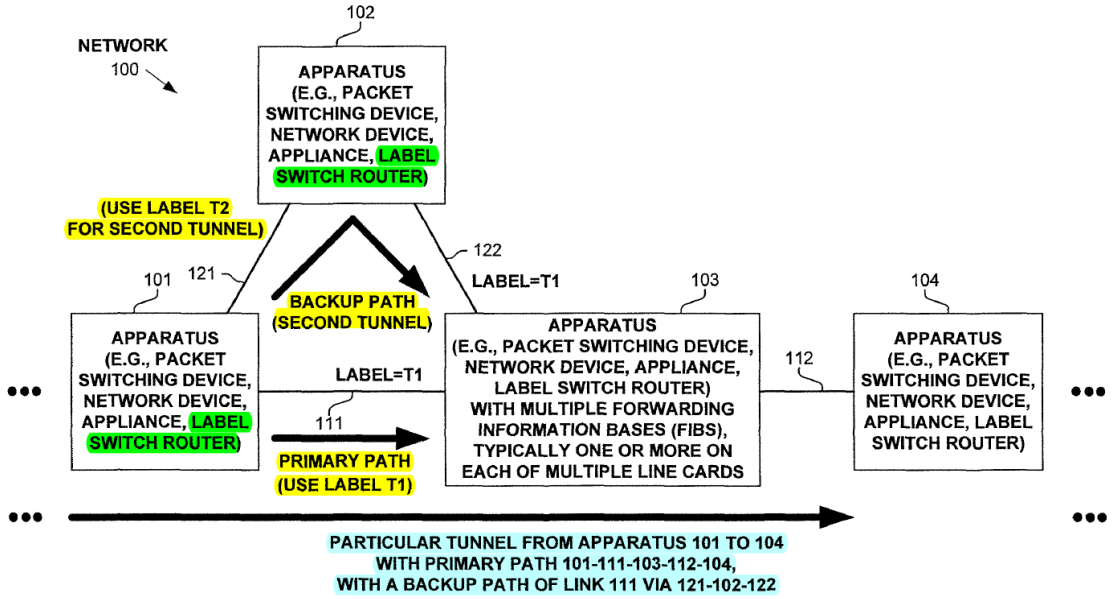
In Figure 4.4, for example, the least-cost path between nodes $A$ (source) and $C$ (destination) is along the path $ADEC$. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)

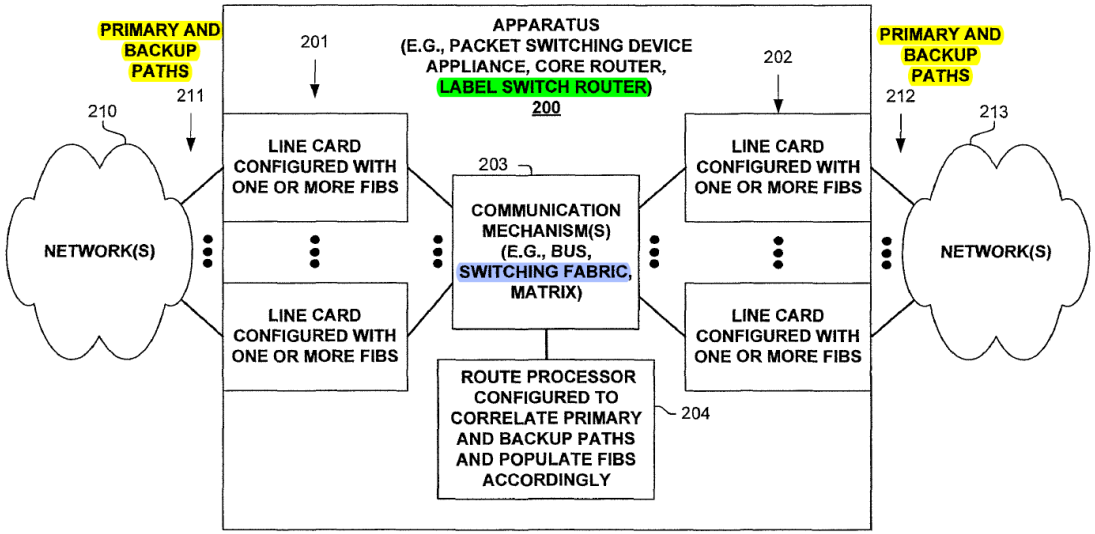
As a simple exercise, try finding the least-cost path from nodes $A$ to $F$, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from $A$ to $F$ by examining Figure 4.4, tracing a few routes from $A$ to $F$, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between $A$ and $F$? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:

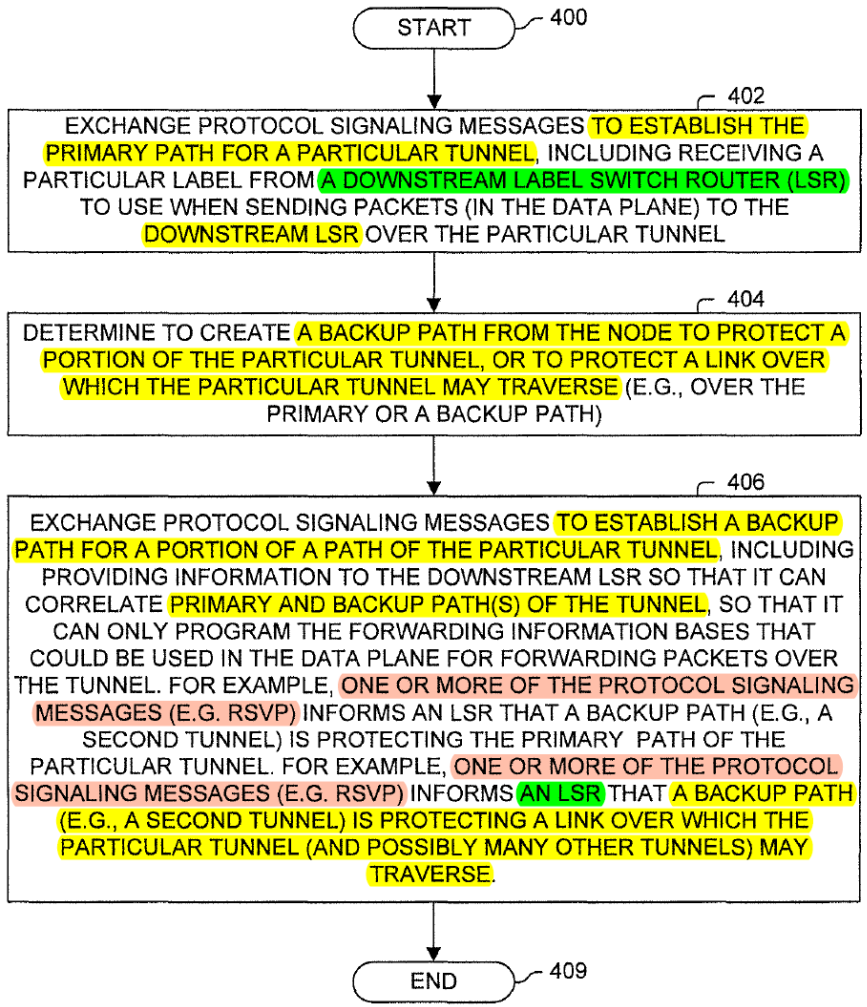
Kurose at 281.

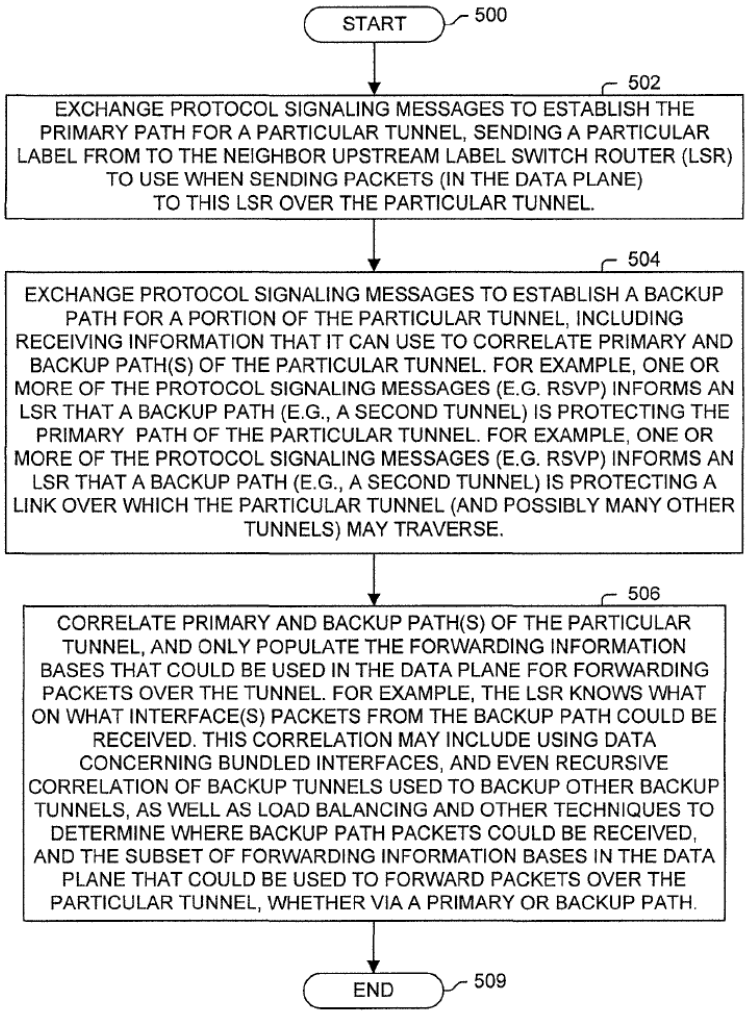
| No. | '821 Patent Claim 10 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| 11 | The method of claim 10, wherein said predefined metric is selected from the group consisting of interior gateway protocol (IGP) and traffic engineering (TE). | The Reference discloses the method of claim 10, wherein said predefined metric is selected from the group consisting of interior gateway protocol (IGP) and traffic engineering (TE).<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Cisco created and developed the MPLS and MPLS-TE standards and patented technology based on those standards *before* Orckit utilized such technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Filsfils<br>• Taylor<br>• Vasseur '879<br>• Rustogi<br><br>**Filsfils discloses**:<br>"In one embodiment, forwarding information bases (FIBs) are selectively populated in a packet switch. A packet switching device determines, based on one or more protocol signaling messages, a subset, which is less than all, on which FIBs a lookup operation may be performed for identifying forwarding information for a received particular packet. The packet switching device populates each of these FIBs, but not all of the FIBs of the packet switching device, with forwarding information corresponding to the particular forwarding |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | value. Thus, FIB resources are consumed for only those FIBs which could actually be used, and not all of the FIBs, for forwarding packets in the data plane of the packet switching device, whether these packets are received on a primary or backup path." Filsfils, Abstract.<br>  <br><br>**FIGURE 1**<br>Filsfils, FIG. 1 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIGURE 2<br>Filsfils, FIG. 2 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 4**<br>Filsfils, FIG. 4 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 5**<br>Filsfils, FIG. 5. |

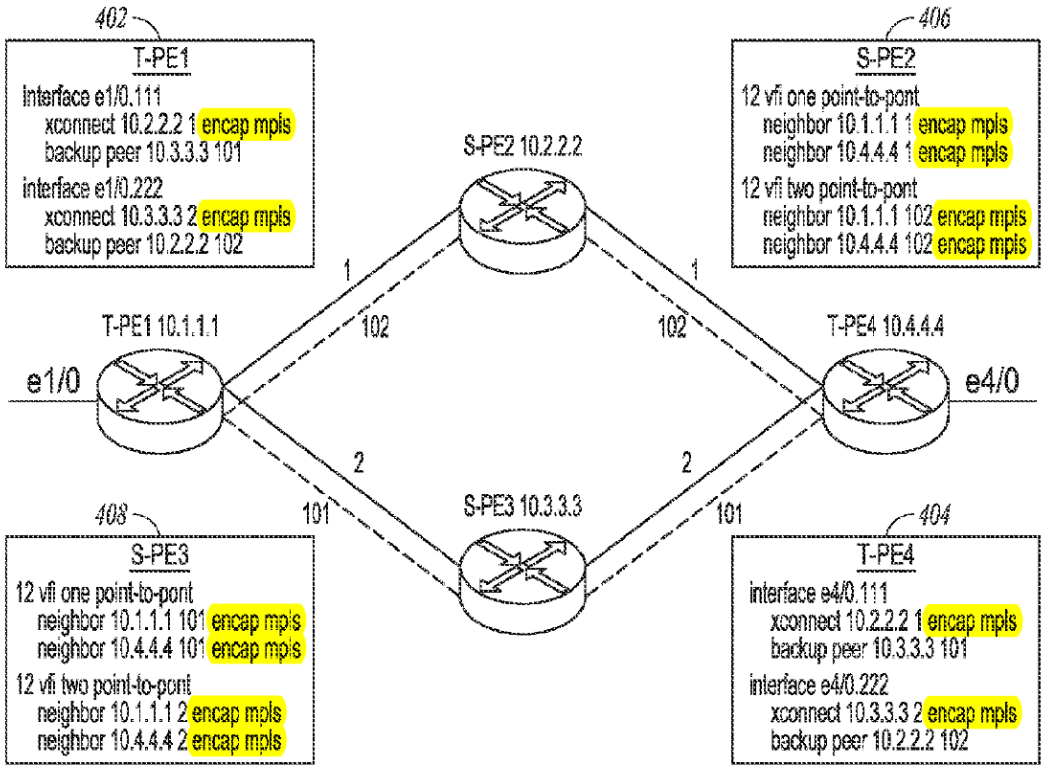
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology, including switching packets on labels especially in the core network using Multiprotocol Label Switching (MPLS)." Filsfils, 1:12-22.<br><br>"Tunnels, such as MPLS-TE (Traffic Engineering) and MPLS-TP (Transport Profile), are paths established through a network in order to transport packets efficiently through a label switched network. Fast Re-Route (FRR) is a technology that allows backup paths to be established in the network, which can be used in case of a problem with a primary path (original primary path or currently used backup path) of the tunnel. RFC 4090, entitled "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," provides an extension of the protocol signaling to establish backup label switched path (LSP) tunnels for local repair of LSP tunnels." Filsfils, 1:23-33.<br><br>"Expressly turning to the figures, FIG. 1 illustrates a network 100 operating according to one embodiment. Shown are four apparatus 101-104 (e.g., packet switching devices such as a label switch router, network device, and/or appliance). For explanation purposes, each of apparatus 101-104 will be referenced as a label switch router (LSR)." Filsfils, 5:41-46.<br><br>"As shown, a particular tunnel is established, using a signaling protocol and exchanging of protocol signaling messages. Note, LSR 101 may, or may not, be an endpoint of the particular tunnel (e.g., LSR 101 may be an intermediate LSR on the path of the particular tunnel). The primary path of the particular tunnel includes spans from LSR 101 via link 111 to LSR 103 and via link 112 to LSR 104. Note, LSR 104 may be an intermediate LSR on the path of the particular tunnel, or an endpoint of the particular tunnel. Further, for this example embodiment, LSR 103 signals LSR 101 to use label T1 at the top of the label stack in the header of a packet sent to it on the particular tunnel." Filsfils, 5:47-58. |

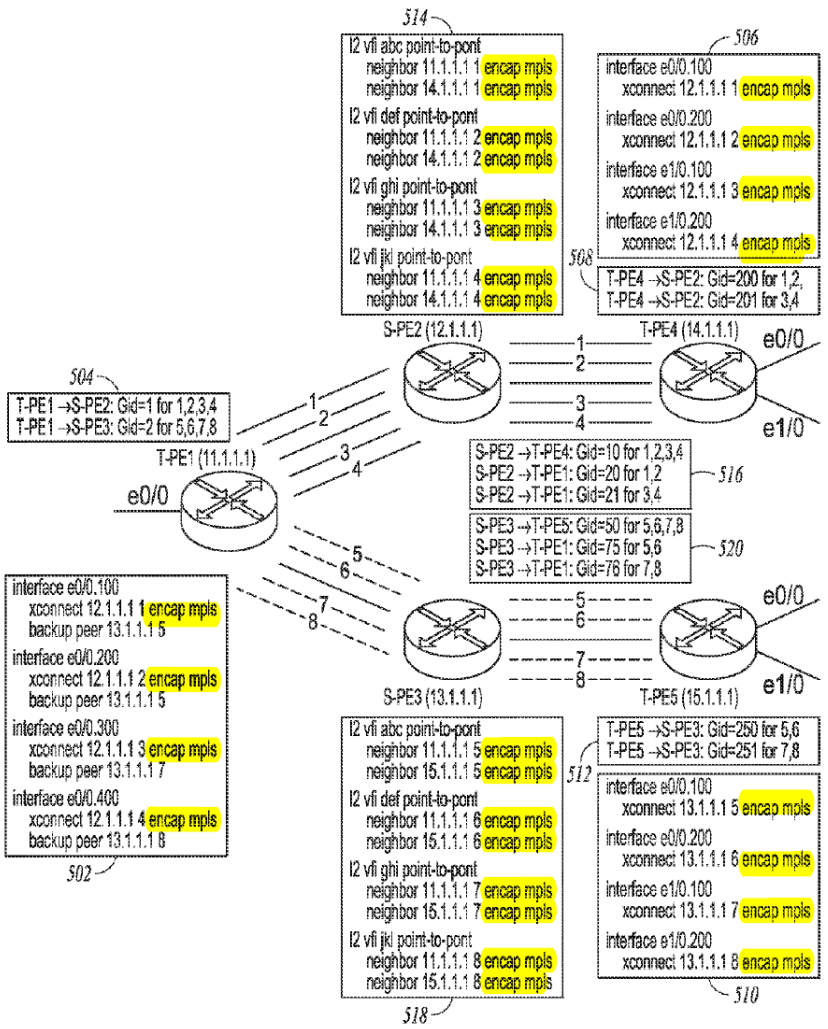
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "A second tunnel from LSR 101 via link 121 to LSR 102 and via link 122 to LSR 103 is similarly configured using a signaling protocol and exchanging of protocol signaling messages. For example purposes, LSR 102 signals LSR 101 to use label (T2) at the top of the label stack in the header of a packet sent to it on the second tunnel. In one embodiment, LSR 101 creates the second tunnel in response to determining, or being instructed to, create a backup path to protect link 111 and/or protect all or certain tunnels traversing link 111." Filsfils, 5:59-67.<br><br>"As shown in FIG. 1, link 111 (primary path of the particular tunnel and/or all or certain tunnels traversing link 111) is protected by LSR 101 using the second tunnel (backup path). When sending packets over the particular tunnel over link 111, LSR 101 includes label T1 at the top of the label stack of these packets. If link 111 cannot be used for communicating packets of the particular tunnel, LSR 101 sends packets over the backup path for the particular tunnel by sending packets to LSR 102, with these packets having a label stack including: label T2 followed by label T1. Thus, LSR 102 will receive these packets, pop the top label (T2) from the label stack of each of these packets, identify based on label T2 to send these packets to LSR 103. After popping the top label, the label at the top of the label stack of these packets is T1, which is the same label LSR 103 expects to receive for the particular tunnel. Therefore, these packets received with label T1 at the top of their label stack, are forwarded (after popping label T1 from their label stack) by LSR 103 over the particular tunnel to LSR 104." Filsfils, 6:6-24.<br><br>"One embodiment acquires such additional information by extending Resource Reservation Protocol (RSVP) to provide information which allows a packet switch to correlate primary and backup paths. Thus, a packet switch can use this additional information in determining which of its forwarding information bases (FIBs) could possibly be used in forwarding packets (e.g., in the data plane of the packet switch)." Filsfils, 6:51-57.<br><br>"In providing this additional information to LSR 103, one embodiment communicates an extended RSVP message (including a new or modified RSVP object) or other message to LSR 103 on the second tunnel. This messages designates one or more primary tunnels (e.g., label T1 in our example) and/or a link (e.g., link 111). As LSR 103 knows what interface |

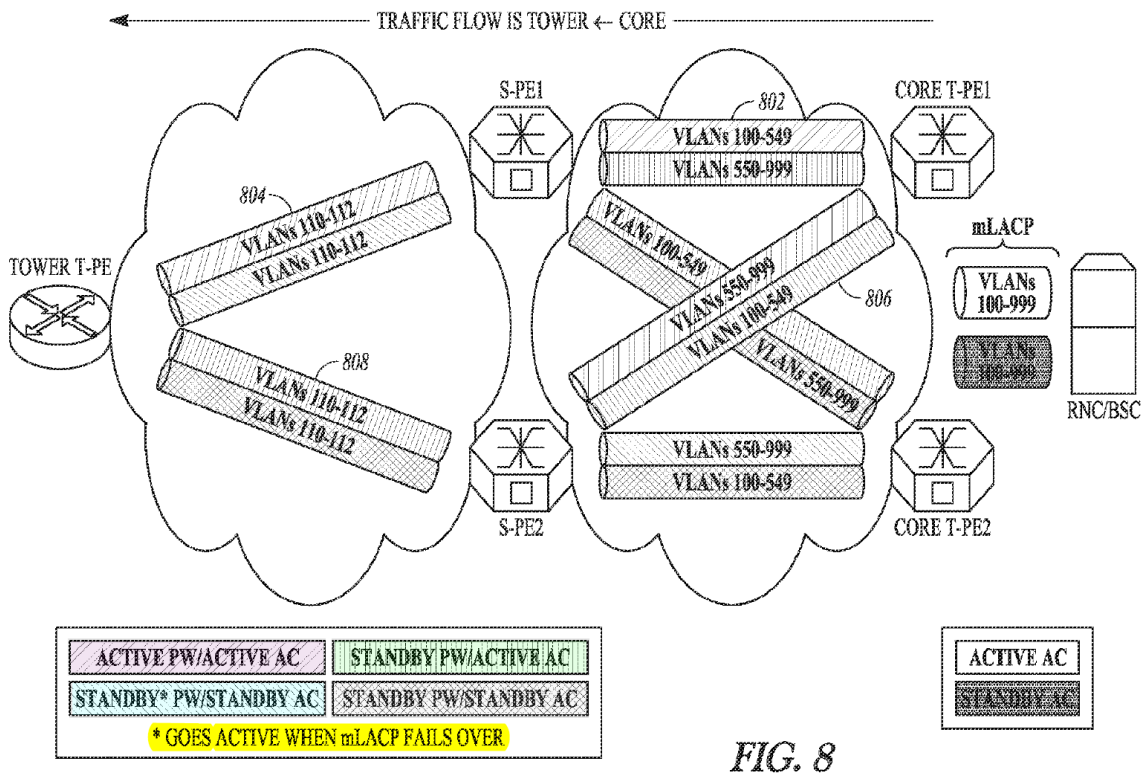
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | that it received this message, LSR 103 knows that it must populate forwarding information for these primary tunnels, either specified (e.g., by a label such as T1), or all labels corresponding to tunnels which could be received over link 111. In one embodiment, the extended RSVP or other message communicated to LSR 103 also includes an identification of the backup tunnel (e.g., T2) over which the RSVP or other message is being received, as the identification the tunnel over which a packet is received is often not communicated in a packet (e.g., in the case of Penultimate Hop Popping)." Filsfils, 7:19-34.<br><br>"As shown in FIG. 1, one embodiment includes apparatus 103, which populates less than all of its FIBs with forwarding information for a tunnel (although all FIBs may be populated for certain tunnels). One embodiment includes apparatus 101 and/or 102 which communicates, via a signaling protocol (e.g., an extension of RSVP, or using another protocol), information which allows apparatus 103 to determine the relationship between primary and backup paths, such that apparatus 104 can correlate this primary and backup path information (possibly also correlating backup path of backup path information, and/or bundled interfaces and/or bundled links) to identify a minimum subset of the FIBs that could possibly be used in forwarding packets of particular primary paths (e.g., tunnels)." Filsfils, 7:49-62<br><br>"Turning to FIG. 2, illustrates an apparatus 200 (e.g., packet switching devices such as a label switch router, network device, and/or appliance) operating in one embodiment. As shown, apparatus 200 includes line cards 201, 202 communicatively coupled via communication mechanism(s) 203 (e.g., bus, switching fabric, and/or matrix). Additionally, route processor 204 is configured to correlate primary and backup paths of tunnels, and to populate minimum subsets of FIBs with forwarding information for labels. Again, a minimum subset of FIBs for a particular path or label of the particular path is the set of FIBs that are determined to possibly be used in forwarding packets of a primary path, whether the label is received in a packet over the primary path or over a backup path, and possibly considering backup paths of a backup path and/or the possibly effect of bundled interfaces and/or bundled links." Filsfils, 7:63-8:11. |

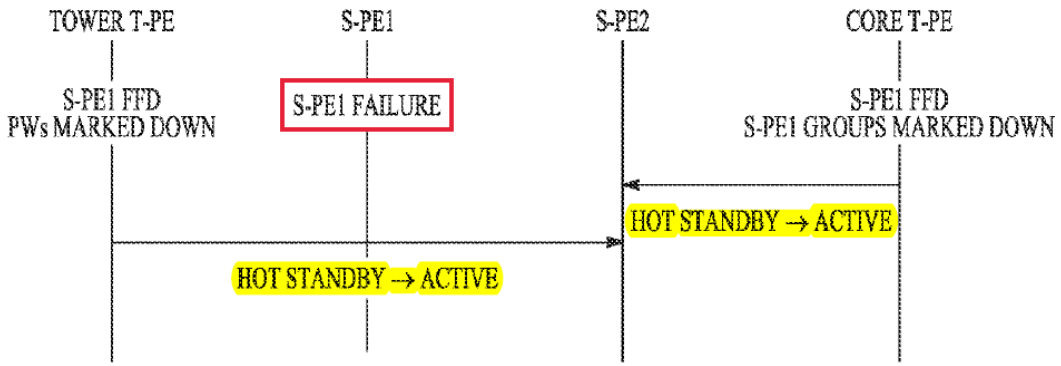
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "As shown in FIG. 2, apparatus 200 is communicatively coupled via primary and backup paths 211, 212 to networks 210 and 213 (which could be the same network). As illustrated, each of line cards 201, 202 includes one or more FIBs. By correlating on which line card(s) 201, 202 and even within line cards 201, 202 that have multiple FIBs, primary and backup path(s) of tunnels, the number of FIB entries populated in apparatus 200 can typically be reduced, possibly significantly saving memory/storage resources and resources used to populate the FIBs." Filsfils, 8:12-21.<br><br>"FIG. 4 illustrates a process performed in one embodiment. Processing begins with process block 400. In process block 402, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes receiving a particular label for a downstream LSR to use when sending packets to the downstream LSR over the particular tunnel." Filsfils, 8:61-67.<br><br>"In process block 404, a determination is made to create a backup path from the node (e.g. the node performing these operations). This backup path may be used to protect one or more particular tunnels, and/or may be used to protect a link which is used to carry packet traffic of one or more tunnels." Filsfils, 9:1-5.<br><br>"In process block 406, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including providing information to the downstream LSR so that the downstream LSR can correlate primary and backup path(s) of the particular tunnel and substantially only program the FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:6-23. |

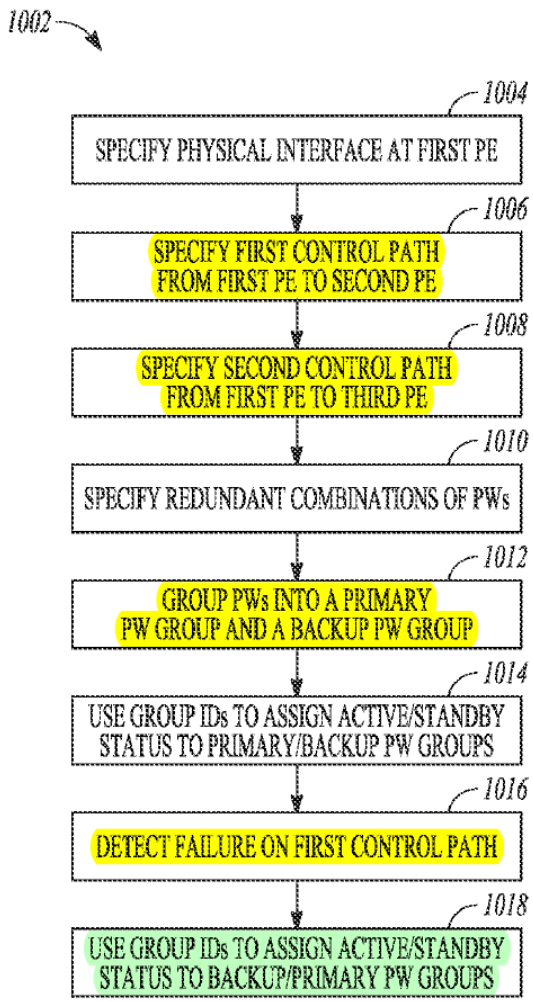
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "FIG. 5 illustrates a process performed in one embodiment. Processing begins with process block 500. In process block 502, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes sending a particular label for an upstream LSR to use when sending packets over the particular tunnel to this apparatus (e.g., an LSR performing these operations)." Filsfils, 9:26-32.<br><br>"In process block 504, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including receiving information that the LSR can use to correlate primary and backup path(s) of the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:33-47.<br><br>"In process block 506, the primary and backup path(s) of the particular tunnel are correlated to identify the set of FIBs that could possibly be used in forwarding packets of the particular tunnel. Substantially only those FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel (either through a primary or backup path) are populated with the forwarding information (e.g., an entry corresponding to the label it advertised to use for the particular tunnel) for the particular tunnel. For example, the LSR knows what on what interface(s) packets from the backup path could be received. This correlation may include using data concerning bundled interfaces, and even recursive correlation of backup tunnels used to backup other backup tunnels, as well as load balancing and other techniques to determine where backup path packets could be received, and the subset of forwarding information bases in the data plane that could be used to forward packets over the tunnel, whether via a primary or backup path." Filsfils, 9:48-65. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | **Taylor discloses**:<br>"Grouping pseudowires based on hardware interfaces and configured control paths enables improved pseudowire failover performance. Signaling status changes (e.g., from standby to active status) is facilitated by using group IDs for the pseudowire groups, thereby enabling improved failover performance when there is disruption in the network." Taylor, Abstract.<br>  <br><br>*FIG. 4*<br>Taylor, FIG. 4 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 5*<br>Taylor, FIG. 5 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br>Taylor, FIG. 8 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Taylor, FIG. 9 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Taylor, FIG. 10 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The present disclosure relates generally to communication networks and more particularly to pseudowire configurations in communication networks." Taylor, 1:8-10.<br><br>"Virtual Private Network (VPN) services provide secure network connections between different locations. A company, for example, can use a VPN to provide secure connections between geographically dispersed sites that need to access the corporate network so that each customer edge (CE) end point or node can communicate directly and independently with all other CE nodes. Different types of VPNs have been classified by the network layer used to establish the connection between the customer and provider network. For example, Virtual Private LAN Service (VPLS) is an architecture that delivers a multipoint Layer 2 VPN (L2VPN) service that in all respects emulates an Ethernet Local Area Network (LAN) across a wide metropolitan geographic area. All services in a VPLS appear to be on the same LAN, regardless of location. In other words, with VPLS, customers can communicate as if they were connected via a private Ethernet segment, i.e., multipoint Ethernet LAN services." Taylor, 1:12-28.<br><br>"In this context, each CE device at a customer site is connected to the service provider network at a provider edge PE) device by an Attachment Circuit (AC) that provides the customer connection to a service provider network, that is, the connection between a CE node and its associated PE node. Within the provider network, each PE device includes a Virtual Switch Instance (VSI) that emulates an Ethernet bridge (i.e., switch) function in terms of Media Access Control (MAC) address learning and forwarding in order to facilitate the provisioning of a multipoint L2VPN. A pseudowire (PW) is a virtual connection between two PE devices that connect two attachment circuits. In the context of the VPLS service, a pseudowire can be thought of as a point-to-point virtual link for each offered service between a pair of VSIs. Therefore, if each VSI can be thought of as a virtual Ethernet switch for a given customer service instance, then each pseudowire can be thought of as a virtual link connecting these virtual switches to each other over a Packet Switched Network (PSN) for that service instance." Taylor, 1:29-47. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Since the failure of pseudowires obviously degrades network performance, some effort has been directed towards adding system redundancies including redundant pseudowires. However, the presence of redundant pseudowires alone is insufficient to improve overall failover performance, that is, the ability to switch over automatically to a redundant or backup system. Thus, there is a need for improved methods for managing pseudowires to facilitate pseudowire switching and enable improved failover performance." Taylor, 1:48-56.<br><br>"FIG. 4 shows details for PW connectivity in an exemplary network for an example embodiment." Taylor, 1:66-67.<br><br>"FIG. 5 shows details for PW grouping for an example embodiment." Taylor, 2:1-2.<br><br>"FIG. 8 shows an example network including redundant PW connections for an example embodiment." Taylor, 2:9-10.<br><br>"FIG. 9 shows an example sequence diagram for a failure mode related to the embodiment shown in FIG. 8." Taylor, 2:11-12.<br><br>"FIG. 10 shows a flowchart that illustrates a method of providing improved PW grouping according to an example embodiment." Taylor, 2:13-15.<br><br>"According to one embodiment, a method of providing improved pseudowire performance includes specifying a physical interface at a first PE node in a network, a first control path from the first PE node to a second PE node in the network, and a second control path from the first PE node to a third PE node in the network. With these specifications, the method then includes specifying redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node, and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Then these pseudowires can be grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | pseudowires. Group identifiers for the primary PW group and the backup PW group can then be used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths. The method may then include detecting a failure on the first control path, and in response to the detected failure, using the group identifiers to assign the active status to the backup pseudowires and the standby status to the primary pseudowires." Taylor, 2:27-52.<br><br>"Pseudowires are used in pseudowire emulation edge-to-edge to provide a Layer 2 Virtual Private Network (L2VPN) connection. When large numbers (e.g., 4,000-6,000) of pseudowires are aggregated together on a single router, failure performance tends to be linear or O(n) where n is the number of pseudowires. While O(n) performance may be acceptable for small numbers of pseudowires, the effect on network outages can be increasingly undesirable as the number of pseudowires increases." Taylor, 2:54-62.<br><br>"For example, a cell-site router will typically start an approximately 2-minute procedure if contact with its controller, which is reached via a pseudowire, is lost for more than some threshold amount (e.g., between approximately 0.75 and 1.75 seconds in some cases). This can be a major impediment to the scalability of pseudowire deployments. These issues have become increasingly relevant as providers of Multiservice Broadband Networks (MBNs) are rapidly replacing or augmenting their traditional Synchronous Optical Networking (SONET) equipment with cheaper Ethernet equipment in the evolution towards a 4G (i.e., 4$^{th}$ generation) network." Taylor, 2:63-3:7.<br><br>"One aspect of a solution to the problem of pseudowire failure is the deployment of redundant pseudowires. For example, redundant pseudowires have been used in the context of Multiprotocol Label Switching (MPLS) networks, which use a Label Distribution Protocol (LDP) to manage labels for forwarding traffic between routers. In this context, general requirements for redundancy schemes have been developed so that duplicate pseudowires are available when a given pseudowire fails (e.g., by using active/standby status indicators). In addition, more specific implementations for redundant pseudowires have also been developed." Taylor, 3:8-18. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "FIG. 1 shows a reference network model 102 with applications to example embodiments disclosed herein. The reference network model 102 includes an aggregation network 104 of PE nodes and a distribution network 106 of PE nodes between a radio network controller (RNC) (or base station controller (BSC)) 108 on the core side of the model 102 and a radio tower 111 on the tower side of the model 102. Switching provider edge nodes S-PE1 and S-PE2 connect the two networks 104, 106. On the core side, two core terminating provider edges T-PE1 and T-PE2 connect to the RNC/BSC 108 through attachment circuits 110, 112. On the tower side, one tower terminating provider edge T-PE connects to the radio tower 111 through an attachment circuit 114." Taylor, 3:19-31.<br><br>"Additionally as noted in FIG. 1, peer-PE monitoring is carried out within each network 104, 106. That is, there is peer-PE monitoring between provider edges that share a segment, for example, by multi-hop bidirectional forwarding detection (BFD). Alternatively, peer monitoring can be accomplished by other means (e.g., MPLS-TP (Transport Protocol) keep-olives). This peer-PE monitoring is used to provide the mechanism for fast failure detection. Once a failure is detected, the network can react by "rerouting" the failed pseudowires to pre-provisioned backup paths and thus provide a minimal disruption in service to the end-user. This rerouting can be accomplished by LDP signaling between provider edges." Taylor, 3:32-44.<br><br>"The reference network model 102 may be considered as part of a larger hub-and-spoke model as shown in FIG. 2. A hub-and-spoke distribution model 202 includes a core network 204, distribution networks 206, and aggregation networks 208. Network elements including distribution nodes, aggregation nodes, and towers are also shown with nominal count values (e.g., 30 distribution nodes between the core network 204 and a distribution network 206). In this model 202, tower T-PEs are the spokes white core-PEs constitute the hub. Dozens to hundreds of tower T-PEs connect to a few S-PEs; these S-PEs are quite similar to ASBRs as they act as forwarders between the two distinct MPLS domains, providing isolation and, in the case of mobility, aggregation services. Typically, several aggregation networks 208 are connected to a single distribution network 206, eventually connecting the tower with the core router that connects the tower's ACs to the RNC/BSC. There are typically several |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | distribution networks in a Radio Access Network (RAN) connected to the service provider's core Internet Protocol (IP) network." Taylor, 4:11-30.<br><br>"With reference to FIG. 1, FIG. 3 shows a variety of failure modes encountered in the reference network model 102. Failure 302 of communications between tower T-PE and the S-PE can be detected via peer monitoring when both the S-PE and the T-PE are still active/alive. For example, this failure may be due to a toss of connectivity when the BFD session goes down. Failure 304 of S-PE1 can be due to a hardware failure, power outage, or the lack of BED-session maintenance capability (e.g., if the BFD-session hello timers cannot be serviced for the prescribed period of time). Failure 306 of communications between S-PE1 and core T-PE1 can be detected via peer monitoring when both S-PE1 and core T-PE1 are still active/alive. Failure 308 at core T-PE1 can be due to a hardware failure, power outage, or the lack of BFD-session maintenance capability." Taylor, 4:46-60.<br><br>"Pseudowire connectivity is further illustrated in FIG. 4 where the illustrated network includes four nodes: T-PE1 (10.1.1.1), S-PE2 (10.2.2.2), S-PE3 (10.3.3.3), and T-PEA (10.4.4.4). For the terminating nodes T-PE1 and T-PE4, specifications for VLANs (virtual Local Area Networks) connections (i.e., pseudowires) are shown using the Internet Operating System Command Line Interface (IOS CLI). The specification 402 for T-PE1 defines two VLANS as primary/backup combinations of virtual circuits for the network. The first three lines of the specification 402 define "VLAN 111" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 111" in the first line. The second line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE2 (10.12.2) with a virtual circuit Identification (VCID) set as VCID=1, and the third line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=101. The next three lines of the specification 402 define "VLAN 222" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 222" in the fourth line. The fifth line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=2, and the sixth line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE2 (10.2.2.2) with VCID=102. These virtual circuits, VCID=1, VCID=2, VCID=101 and VCID=102 are shown in the figure between T-PE1 and the S-PEs with a |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | solid line for the primary circuits VCID=1 and VCID=2 and a dashed line for the backup circuits VCID=101 and VCID=102." Taylor, 4:61-5:23.<br><br>"With respect to T-PE1 in FIG. 4, although "VLAN 111" and "VLAN 222" share the same hardware port, they do not share the same "control path disposition." That is, "VLAN 111" is primary to S-PE2 (VCID=1) and standby to S-PE3 (VCID=101), white "VLAN 222" has an opposite configuration since it is primary to S-PE3 (VCID=2) and standby to S-PE2 (VCID=102). As discussed below, certain embodiments group pseudowires according to "control path disposition" (e.g., xconnect configuration as well as the hardware interface in order to improve failover performance. That is, to deal with both hardware port failures and switching path failures, the grouping criteria also considers the cross connects. In this case, on T-PE1 as well as T-PE4, there would exist two groups: one for active to S-PE2 and standby to S-PE3 and another for active to S-PE3 and standby to S-PE2 (i.e., the inverse configuration)." Taylor, 5:61-6:9.<br><br>"First, local connectivity is characterized by local group identifications (Group-IDs), which depend on whether the allocation is done at a T-PE or S-PE. FIG. 5 shows an embodiment that illustrates an allocation of local group IDs in a network including terminating nodes T-PE1 (11.1.1.1), T-PE4 (14.1.1.1), and T-PE5 (15.1.1.1) and switching nodes S-PE2 (12.1.1.1) and S-PE3 (13.1.1.1). The specification 502 for T-PE1 determines corresponding local group IDs 504 based on the hardware interface and the control path. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, and the Group-ID=2 corresponds to VCID=5, VCID=6, VCID=7, and VCID=8." Taylor, 6:10-21.<br><br>"Local group IDs are maintained in a database so that pseudowire redundancy is also maintained. First, in a case without pseudowire redundancy, all the xconnect configurations from the same physical interface to the same peer are assigned the same local group ID. So, for example, in Ethernet cases all xconnect configurations under sub-interfaces of the same physical interface to the same peer will be assigned the same local group ID (e.g., e0/0 and e0/1 are sub-interfaces of e0). FIG. 6 shows a database representation for T-PE4 from FIG. 5. From the root node 602 for T-PE4, there is a first interface node 604 for e0/1 and a second interface node 606 for e1/0. The first interface node 604 is configured towards a single peer |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | node (12.1.1.1) 608 and is thus assigned a single local group ID (Group-ID=200) 610. Similarly, the second interface node 606 is configured towards a single peer node (12.1.1.1) 612 and is thus assigned a single local group ID (Group-ID=201) 614. In this case from the assignment of local group IDs 508 in FIG. 5, Group-ID=200 corresponds to VCID=1 and VCID=2, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). Both pseudowires (VCID=1 and VCID=2) are assigned the same local group ID (Group-ID=200) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly, from the assignment of local group IDs 508 in FIG. 5, T-PE4 has VCID=3 and VCID=4 under the physical interface e1/0 going to the same peer S-PE2 (12.1.1.1), and the local Group ID (Group-ID=201) is assigned to these VCs." Taylor, 6:48-7:8.<br><br>"For the pseudowire redundancy case, a separate redundancy-group database is maintained by the xconnect application. This redundancy-group database contains the peer IDs in the group and the local group IDs advertised to them. This is needed to maintain a 1:1 mapping between the primary pseudowires and their corresponding backup pseudowires. FIG. 7 shows a database representation for T-PE1 from FIG. 5. From the root node 702 for T-PE1, there is an interface node 704 for e0/0 and a redundancy group node 706 that shows connections for configurations to a first peer node (12.1.1.1) 708, which is assigned a local group ID (Group-ID=1) 710, and a second peer node (13.1.1.1) 712, which is assigned a local group ID (Group-ID=2) 714. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). These pseudowires are assigned the same group ID (Group-ID=1) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly from the assignment of local group IDs 504 in FIG. 5, T-PE1 has VCID=5, VCID=6, VCID=7, and VCID=8 under the physical interface e0/0 going to another peer S-PE3 (13.1.1.1), and the local group ID (Group-ID=2) is assigned to these VCs. In this case these local group IDs are organized as a redundancy group 706." Taylor, 7:9-35. |

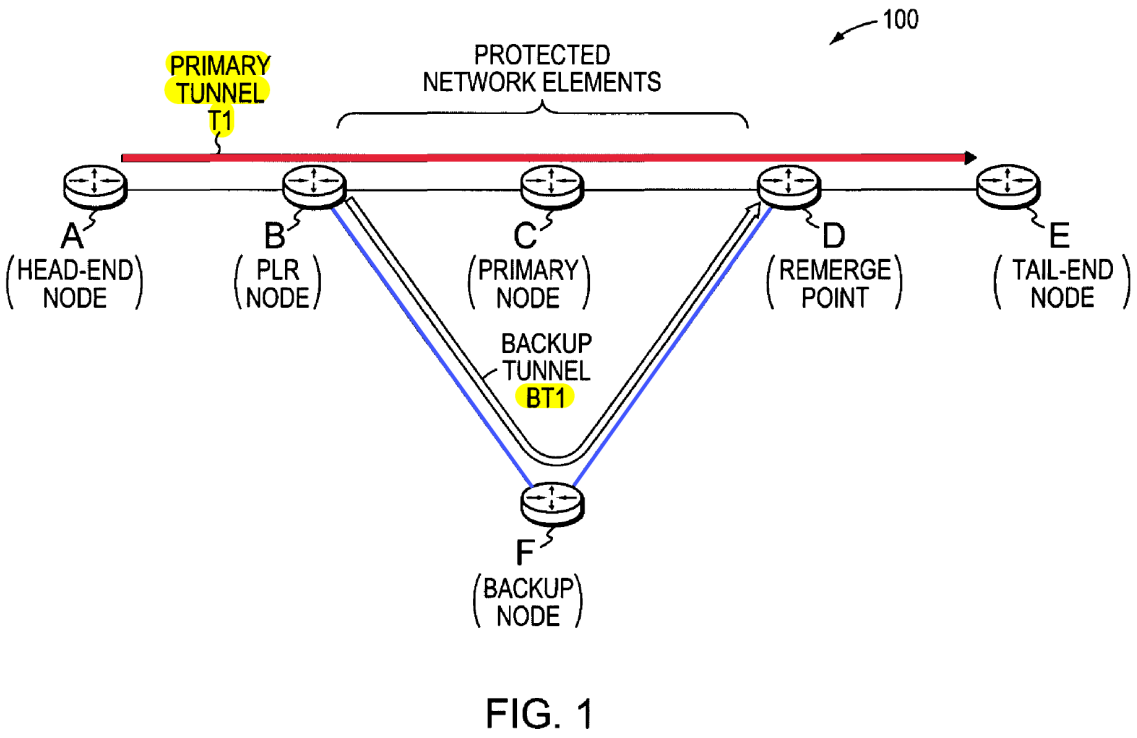
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Pseudowire grouping allows multiple pseudowires to be aggregated together when signaling either label withdrawals or status changes between segment end-point provider edges. This signaling can be carried out through LDP grouping TLV (Type Length Value). For example, when a PE node fails (e.g., failure 304 of S-PE1), aggregating the inter-segment PE signaling using the grouping TLV can provide significant scaling advantages. This allows all pseudowires sharing a physical port and PW configuration (e.g., xconnect configuration) to be signaled en masse between segment-adjacent provider edges." Taylor, 7:36-46.<br><br>"FIG. 8 shows an example based on FIG. 1 where VLAN ACs are shown as grouped by both port/HW-interface and pseudowire-class. The grouping criterion allows all "similar" pseudowires to be signaled together: All the grouped pseudowires share the same port and next-hop provider edge. Additionally, the figure contains many pseudowires, each grouped into a shaded tube. For example, the tube labeled "VLANs 100-549" contains 450 pseudowires grouped together. This figure depicts an incoming Ethernet comprised of 900 VLANs being segmented in two with 450 VLANs (100-549) active to S-PE1 while the other half of the VLANs (550-999) being active to S-PE2. This might be considered a type of manual load balancing. Furthermore, the aggregation network is only showing a single tower and the VLANs associated with it; other VLAN destinations are not shown in the figure." Taylor, 7:47-62.<br><br>"'VLANs 110-112' are active along a first pseudowire path 802 from Core T-PE1 to S-PE1 and a second pseudowire path 804 from S-PE1 to Tower T-PE. When a failure occurs at S-PE1 (e.g., as the switching node failure 304 shown in FIG. 3), then the standby pseudowires become active for 'VLANs 110-112' along a first pseudowire path 806 from Core T-PE1 to S-PE2 and a second pseudowire path 808 from S-PE2 to Tower T-PE." Taylor, 7:63-8:3.<br><br>"The standby pseudowires in FIG. 8 can be configured as HSPWs, a configuration that enables ACs to quickly failover to pre-provisioned pseudowires that are in active state but set to not-forwarding. Then when a failure occurs, switching over to these pre-provisioned HSPWs occurs quickly by switching from not-forwarding status to forwarding status." Taylor, 8:4-9. |

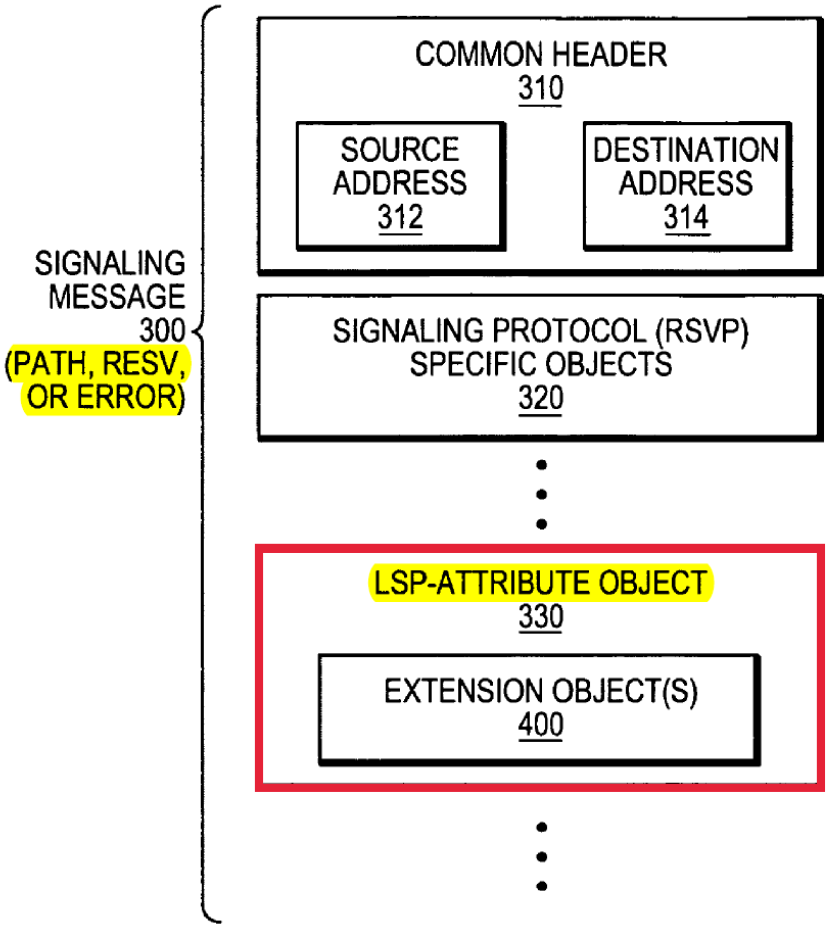
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "FIG. 9 shows a UML (Unified Modeling Language) sequence diagram of signaling events related to the failover procedure illustrated in FIG. 8 for a switching node failure 304. The Tower T-PE and the Core T-PE independently detect a failure at S-PE1 (e.g., BFD forwarding with LDP signaling), and then mark down the status of the currently active pseudowires routed through S-PE1 and mark up the status of the standby pseudowires routed through S-PW2. Other failure modes shown in FIG. 3 can be handled similarly." Taylor, 8:10-18.<br><br>"In general, it is desirable for MPLS-based. Ethernet networks to react quickly to failures, so proactive detection mechanisms are employed in order to pick up system failures quickly. All proactive monitoring is typically done between PE peers on a single MPLS network. These provider edges on the edges of the MPLS network act similarly to an Autonomous System Boundary Router (ASBR). As a result, related embodiments detect control path failures, which may not be the same as pseudowire data path failures. That is, the data packets and control packets may take different paths between provider edges in a MPLS network although typically these paths are coincident. Thus, when the control and data paths are not coincident, if the control path fails, then all pseudowires utilizing the control path are marked as failed. As a corollary, if the data path fails and the control path remains healthy, then failure will not be detected from monitoring the control path." Taylor, 8:19-35.<br><br>"A failure of a monitored provider edge initiates a switchover of all active pseudowires using the failing provider edge to their configured HSPWs (if they exist). Grouping can greatly reduce the number of messages needed between provider edges (Inter-PE Aggregation) and within a single provider edge (Intra-PE Aggregation). Furthermore, the MPLS network itself may be internally resilient deploying technologies such as, but not limited to, MPLS-TE (MPLS Traffic Engineering) and ERR (Fast Reroute). The paths across the MPLS network may recover quickly and might not trip the fault-monitoring systems." Taylor, 8:36-46.<br><br>"With reference to the above discussion, FIG. 10 shows a method 1002 of providing improved PW grouping according to an example embodiment. In a first operation 1004 of the method 1002, a physical interface is specified at a first PE node in a network. In a second |

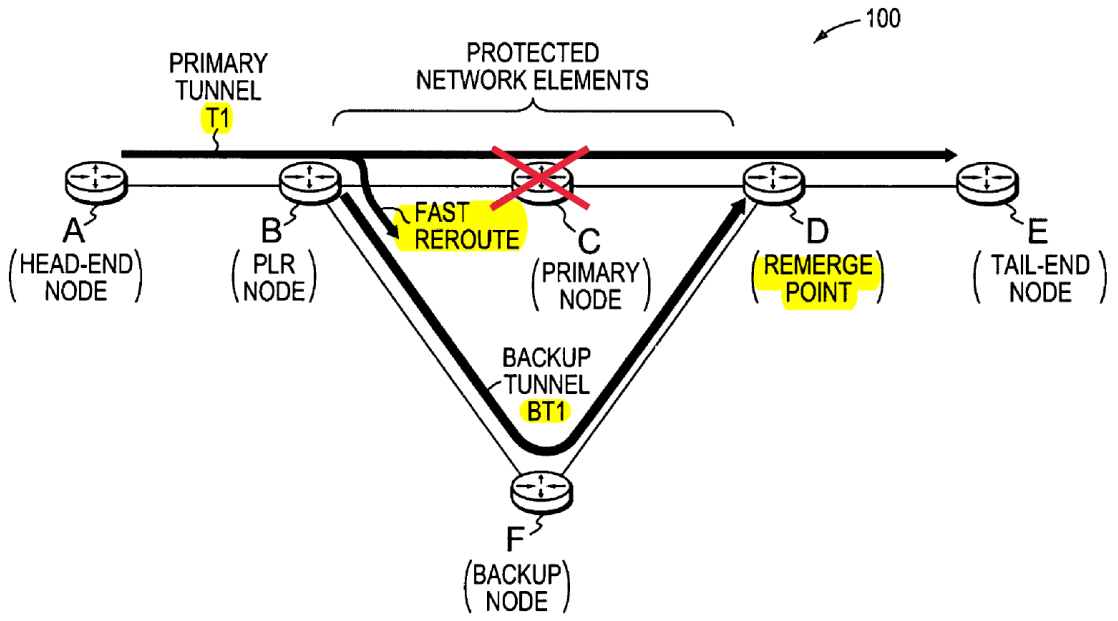
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | operation 1006, a first control path is specified from the first PE node to a second PE node in the network. In a third operation 1008, a second control path is specified from the first PE node to a third PE node in the network. These control paths related to a common physical interface can be used to characterize redundant pairs of pseudowires." Taylor, 8:48-58.<br><br>"In a fourth operation 1010, redundant combinations of pseudowires are specified, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Although a redundant combination may relate a single backup pseudowire to a given primary pseudowire, in some cases multiple backup pseudowires will be related to a given primary pseudowire for increased redundancy. In a fifth operation 1012, these pseudowires are grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. In a sixth operation 1014, group identifiers for the primary PW group and the backup PW group are used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 8:59-9:11.<br><br>"In an optional seventh operation 1016, a failure may be detected on the first control path, and in an optional eighth operation 1018, in response to the detected failure, the group identifiers may be used to assign the active status to the backup pseudowires and the standby status to the primary pseudowires. For example, the failure on the first control path may be detected by using BED packet streams between PE nodes of the network. Then the detected failure can be signaled to PE nodes in the network by sending LDP status updates between PE nodes in the network. Then, after receiving the failure detection signals, the group identifiers can be used again to assign the active status to the backup pseudowires and the standby status to the primary pseudowires by sending LDP status updates between PE nodes in the network." Taylor, 9:12-26. |

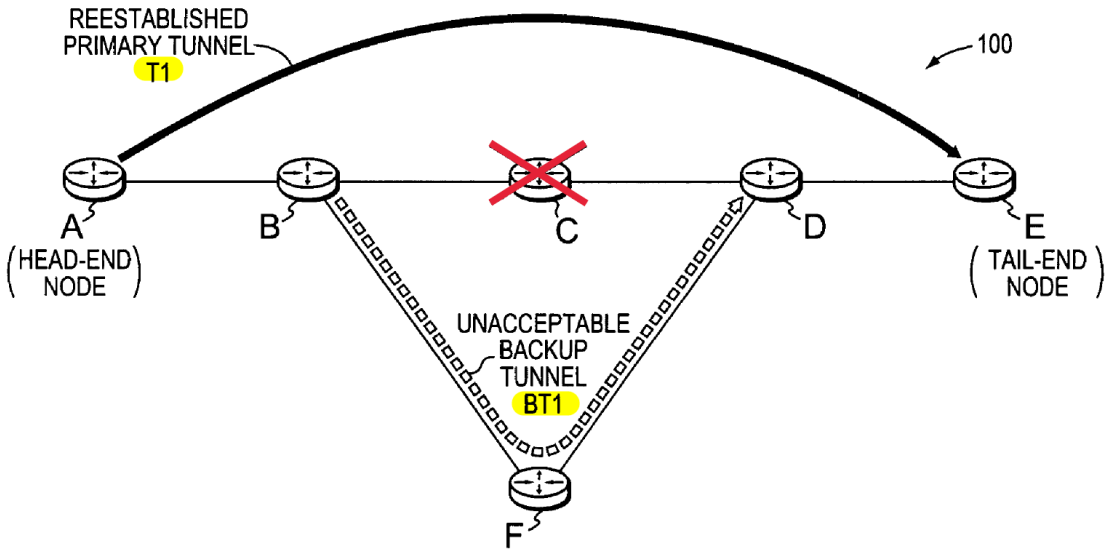
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Typically the network in is an MPLS network and the PE nodes are routers that provide network services to connected CE nodes of a customer network. In general, each control path is an Internet Protocol (IP) routing path between PE nodes in the network and each data path is a label switched path (LSP) between PE nodes in the network." Taylor, 9:27-32.<br><br>"FIG. 11 shows a schematic representation of an apparatus 1102, in accordance with an example embodiment. For example, the apparatus 1102 may be used to implement the method 1002 of providing improved pseudowire grouping as described above with reference to FIG. 10. The apparatus 1102 is shown to include a processing system 1104 that may be implemented on a server, client, or other processing device that includes an operating system 1106 for executing software instructions." Taylor, 10:2-10.<br><br>"In accordance with an example embodiment, the apparatus 1102 includes a PW management module 1108 that includes a first specification module 1110, a second specification module 1112, third specification module 1114, a fourth specification module 1116, a grouping module 1118, and an assignment module 1120. The first specification module 1110 operates to specify a physical interface at a first PE node in a network. The second specification module 1112 operates to specify a first control path from the first PE node to a second PE node in the network. The third specification module 1114 operates to specify a second control path from the first PE node to a third PE node in the network. The fourth specification module 1116 operates to specify redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node." Taylor, 10:11-29.<br><br>"The grouping module 1118 operates to group the pseudowires into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. The assignment module 1120 operates to use group identifiers for the PW groups to assign an active status to the primary pseudowires and a standby status to the backup pseudowires. The active status enables data transfers along corresponding PW data |

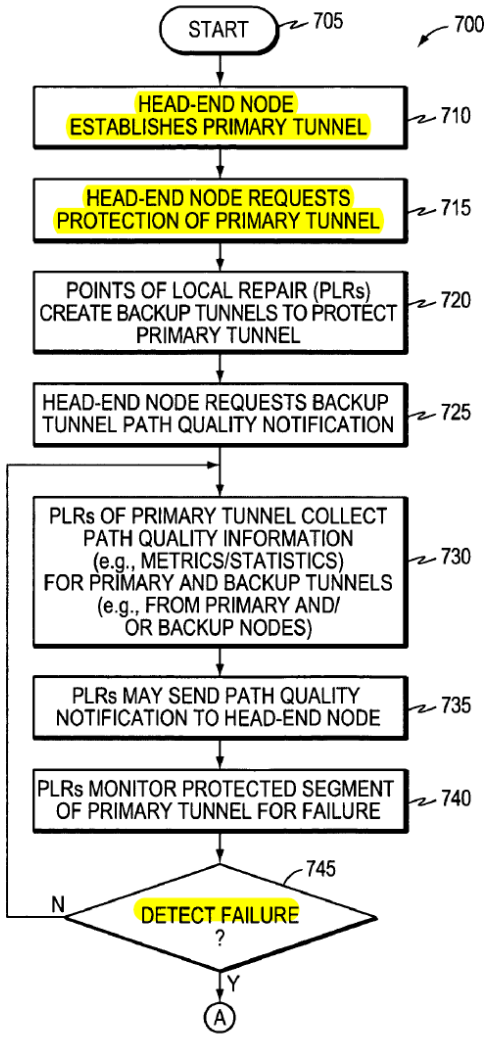
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 10:30-38.<br><br>**Vasseur '879 discloses**:<br>"A technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, Abstract. |

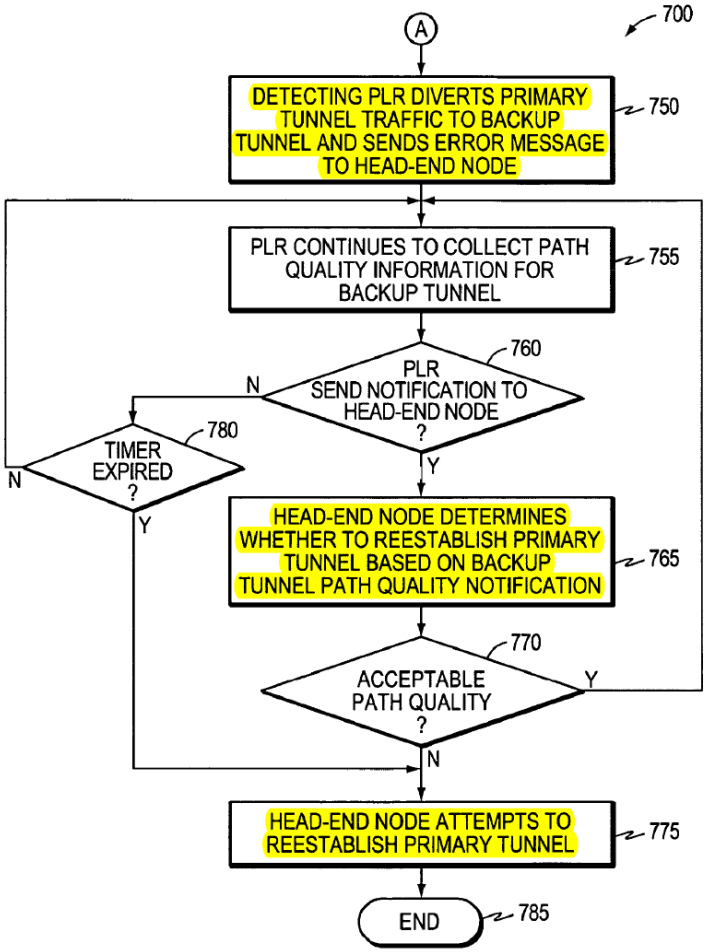
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br>Vasseur '879, FIG. 1 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Vasseur '879, FIG. 3 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Vasseur '879, FIG. 5 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br>Vasseur '879, FIG. 6 (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 7A<br>Vasseur '879, FIG. 7A (annotated). |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 7B<br><br>Vasseur '879, FIG. 7B. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "The present invention relates to computer networks and more particularly to dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 1:10-14.<br><br>"Since management of interconnected computer networks can prove burdensome, smaller groups of computer networks may be maintained as routing domains or autonomous systems. The networks within an autonomous system (AS) are typically coupled together by conventional "intradomain" routers configured to execute intradomain routing protocols, and are generally subject to a common authority. To improve routing scalability, a service provider (e.g., an ISP) may divide an AS into multiple "areas." It may be desirable, however, to increase the number of nodes capable of exchanging data; in this case, interdomain routers executing interdomain routing protocols are used to interconnect nodes of the various ASes. Moreover, it may be desirable to interconnect various ASes that operate under different administrative domains. As used herein, an AS or an area is generally referred to as a "domain," and a router that interconnects different domains together is generally referred to as a 'border router.'" Vasseur '879, 1:40-56.<br><br>"An example of an interdomain routing protocol is the Border Gateway Protocol version 4 (BGP), which performs routing between domains (ASes) by exchanging routing and reachability information among neighboring interdomain routers of the systems. An adjacency is a relationship formed between selected neighboring (peer) routers for the purpose of exchanging routing information messages and abstracting the network topology. The routing information exchanged by BGP peer routers typically includes destination address prefixes, i.e., the portions of destination addresses used by the routing protocol to render routing ("next hop") decisions. Examples of such destination addresses include IP version 4 (IPv4) and version 6 (IPv6) addresses. BGP generally operates over a reliable transport protocol, such as TCP, to establish a TCP connection/session. The BGP protocol is well known and generally described in Request for Comments (RFC) 1771, entitled A Border Gateway Protocol 4 (BGP-4), published March 1995." Vasseur '879, 1:57-2:7. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Examples of an intradomain routing protocol, or an interior gateway protocol (IGP), are the Open Shortest Path First (OSPF) routing protocol and the Intermediate-System-to-Intermediate-System (IS-IS) routing protocol. The OSPF and IS-IS protocols are based on link-state technology and, therefore, are commonly referred to as link-state routing protocols. Link-state protocols define the manner with which routing information and network-topology information are exchanged and processed in a domain. This information is generally directed to an intradomain router's local state (e.g., the router's usable interfaces and reachable neighbors or adjacencies). The OSPF protocol is described in RFC 2328, entitled OSPF Version 2, dated April 1998 and the IS-IS protocol used in the context of IP is described in RFC 1195, entitled Use of OSI IS-IS for routing in TCP/IP and Dual Environments, dated December 1990, both of which are hereby incorporated by reference." Vasseur '879, 2:8-24.<br><br>"An intermediate network node often stores its routing information in a routing table maintained and managed by a routing information base (RIB). The routing table is a searchable data structure in which network addresses are mapped to their associated routing information. However, those skilled in the art will understand that the routing table need not be organized as a table, and alternatively may be another type of searchable data structure. Although the intermediate network node's routing table may be configured with a predetermined set of routing information, the node also may dynamically acquire ("learn") network routing information as it sends and receives data packets. When a packet is received at the intermediate network node, the packet's destination address (e.g., stored in a header of the packet) may be used to identify a routing table entry containing routing information associated with the received packet. Among other things, the packet's routing information indicates the packet's next-hop address." Vasseur '879, 2:25-41.<br><br>"Multi-Protocol Label Switching (MPLS) Traffic Engineering has been developed to meet data networking requirements such as guaranteed available bandwidth or fast restoration. MPLS Traffic Engineering exploits modem label switching techniques to build guaranteed bandwidth end-to-end tunnels through an IP/MPLS network of label switched routers (LSRs). These tunnels are a type of label switched path (LSP) and thus are generally referred to as MPLS Traffic Engineering (TE) LSPs. Examples of MPLS TE can be found in RFC |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels dated December 2001, RFC 3784 entitled Intermediate-System-to-Intermediate-System (IS-IS) Extensions for Traffic Engineering (TE) dated June 2004, and RFC 3630, entitled Traffic Engineering (TE) Extensions to OSPF Version 2 dated September 2003, the contents of all of which are hereby incorporated by reference in their entirety." Vasseur '879, 2:58-3:6.<br><br>"Establishment of an MPLS TE-LSP from a head-end LSR to a tail-end LSR involves computation of a path through a network of LSRs. Optimally, the computed path is the "shortest" path, as measured in some metric, that satisfies all relevant LSP Traffic Engineering constraints such as e.g., required bandwidth, "affinities" (administrative constraints to avoid or include certain links), etc. Path computation can either be performed by the head-end LSR or by some other entity operating as a path computation element (PCE) not co-located on the head-end LSR. The head-end LSR (or a PCE) exploits its knowledge of network topology and resources available on each link to perform the path computation according to the LSP Traffic Engineering constraints. Various path computation methodologies are available including CSPF (constrained shortest path first). MPLS TE-LSPs can be configured within a single domain, e.g., area, level, or AS, or may also span multiple domains, e.g., areas, levels, or ASes." Vasseur '879, 3:7-24.<br><br>"The PCE is an entity having the capability to compute paths between any nodes of which the PCE is aware in an AS or area. PCEs are especially useful in that they are more cognizant of network traffic and path selection within their AS or area, and thus may be used for more optimal path computation. A head-end LSR may further operate as a path computation client (PCC) configured to send a path computation request to the PCE, and receive a response with the computed path, potentially taking into consideration other path computation requests from other PCCs. It is important to note that when one PCE sends a request to another PCE, it acts as a PCC." Vasseur '879, 3:25-36.<br><br>"Some applications may incorporate unidirectional data flows configured to transfer time-sensitive traffic from a source (sender) in a computer network to a destination (receiver) in the network in accordance with a certain "quality of service" (QoS). Here, network resources may be reserved for the unidirectional flow to ensure that the QoS associated with the data |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | flow is maintained. The Resource ReSerVation Protocol (RSVP) is a network-control protocol that enables applications to reserve resources in order to obtain special QoS for their data flows. RSVP works in conjunction with routing protocols to, e.g., reserve resources for a data flow in a computer network in order to establish a level of QoS required by the data flow. RSVP is defined in R. Braden, et al., Resource ReSerVation Protocol (RSVP), RFC 2205, the contents of which are hereby incorporated by reference in its entirety. In the case of traffic engineering applications, RSVP signaling (with Traffic Engineering extensions) is used to establish a TE-LSP and to convey various TE-LSP attributes to routers, such as border routers, along the TE-LSP obeying the set of required constraints whose path may have been computed by various means." Vasseur '879, 3:37-57.<br><br>"Generally, a tunnel is a logical structure that encapsulates a packet (a header and data) of one protocol inside a data field of another protocol packet with a new header. In this manner, the encapsulated data may be transmitted through networks that it would otherwise not be capable of traversing. More importantly, a tunnel creates a transparent virtual network link between two network nodes that is generally unaffected by physical network links or devices (i.e., the physical network links or devices merely forward the encapsulated packet based on the new header). While one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 3:58-4:3.<br><br>"Occasionally, a network element (e.g., a node or link) will fail, causing redirection of the traffic that originally traversed the failed network element to other network elements that bypass the failure. Generally, notice of this failure is relayed to the nodes in the network through an advertisement of the new network topology, e.g., an IGP or BGP Advertisement, and routing tables are updated to avoid the failure accordingly. Reconfiguring a network in response to a network element failure using, e.g., pure IP rerouting, can be time consuming. Many recovery techniques, however, are available to provide fast recovery and/or network configuration in the event of a network element failure, including, inter alia, "Fast Reroute", e.g., MPLS TE Fast Reroute. An example of MPLS TE Fast Reroute is described in Pan, et al., Fast Reroute Extensions to RSVP-TE for LSP Tunnels, RFC 4090, May 2005, which is hereby incorporated by reference as though fully set forth herein." Vasseur '879, 4:4-21. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Fast Reroute (or FRR) has been widely deployed to protect against network element failures, where "backup tunnels" are created to bypass one or more protected network elements (e.g., links, shared risk link groups (SRLGs), and nodes). When the network element fails, traffic is quickly diverted ("Fast Rerouted") over a backup tunnel to bypass the failed element, or more particularly, in the case of MPLS, a set of primary TE-LSPs (tunnels) is quickly diverted. Specifically, the point of local repair (PLR) node configured to reroute the traffic inserts ("pushes") a new label for the backup tunnel, and the traffic is diverted accordingly. Once the failed element is bypassed, the backup tunnel label is removed ("popped"), and the traffic is routed along the original path according to the next label (e.g., that of the original TE-LSP). Notably, the backup tunnel, in addition to bypassing the failed element along a protected primary TE-LSP, also intersects the primary TE-LSP, i.e., it begins and ends at nodes along the protected primary TE-LSP." Vasseur '879, 4:22-39.<br><br>"To offer maximum protection, e.g., guaranteed bandwidth, during Fast Reroute, backup tunnels may reserve a configurable amount of bandwidth to ensure that Fast Rerouted traffic from the primary tunnel has a reserved path to follow. For example, the bandwidth reserved for the primary tunnel may also be reserved on the backup tunnel. While this approach provides maximum protection, it also requires a non-negligible amount of network resources (e.g., capacity/bandwidth) and may increase operational complexity." Vasseur '879, 4:40-48.<br><br>"Certain techniques are available to efficiently minimize the amount of resources required by the establishment and maintenance of the backup tunnels for Fast Reroute. One such technique is to create zero-bandwidth ("0-BW") backup tunnels (i.e., tunnels that reserve no bandwidth) to protect non-0-BW primary tunnels. This "best effort" approach does not guarantee that the path followed by the backup tunnel will have enough bandwidth to support the diverted primary tunnel at the time of failure without QoS degradation, however in many situations the path quality of the backup tunnel is sufficient. For instance, if the network is not overly congested, or the backup tunnel follows a non-congested path, there may be enough available bandwidth along the backup tunnel to support the newly rerouted traffic. Also, because primary tunnels often reserve bandwidth in response to "peak" traffic |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | utilization, the amount of traffic over the primary tunnel at the time of failure may be far less than the reserved bandwidth (e.g., at "off-peak" times). Notably, those skilled in the art will understand that in the absence of the above exceptions, a 0-BW backup tunnel may have unacceptable bandwidth (e.g., affecting path quality) to support the diverted traffic." Vasseur '879, 4:49-5:2.<br><br>"Currently, head-end nodes (LSRs) may be configured to systematically reroute the primary tunnels affected by the network element failure (e.g., diverted primary tunnels), especially in the case with 0-BW backup tunnels, such as, e.g., by reestablishing a new primary tunnel that follows a path excluding the failed network element. In particular, 0-BW backup tunnels represent a best effort attempt to allow the head-end node to more gracefully reestablish the primary tunnel in response to a failure, since the backup tunnels may not be able to support the diverted traffic without QoS degradation. The systematic reestablishing may potentially result in the reestablishment of a large number of primary tunnels (e.g., up to 3000 for a single network element failure in today's networks). Notably, reestablishing diverted primary tunnels may be undesirable for the network, such as by creating traffic churn, jitter, control plane overloads, etc., as will be understood by those skilled in the art. However, as noted above, there are situations where the backup tunnel may provide acceptable bandwidth, at least, for example, for a period of time (e.g., possibly short) until the failed network element is restored. In these situations, then, it may have been unnecessary to reestablish the diverted primary tunnels. There remains a need, therefore, for a technique that dynamically determines whether to reestablish a diverted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 5:3-28.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 5:32-47.<br><br>"In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as Resource ReSerVation Protocol (RSVP) Traffic Engineering (TE) signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or type/length/value (TLV) encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object of the RSVP messages to convey the path quality notification." Vasseur '879, 5:48-58.<br><br>"In accordance with one aspect of the present invention, the head-end node requests the establishment of the primary tunnel (e.g., a TE-Label Switched Path, TE-LSP), along with a request for Fast Reroute protection of one or more network elements (e.g., with zero-bandwidth, 0-BW backup tunnels) at a PLR node. In addition, the head-end node may include a request for backup tunnel path quality notification, such as, e.g., through the use of the novel Feedback sub-object. The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and sends an error message (e.g., an RSVP PathErr) to the head end node indicating the "Fast Rerouting" of the primary tunnel." Vasseur '879, 5:59-6:6.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, the PLR node may collect metrics/statistics (e.g., packet drops, path cost, jitter, etc.) of the primary and/or backup tunnels. Once the primary tunnel is Fast Rerouted, the PLR node continues to collect metrics of the backup tunnel, and may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. Notably, the PLR node may be configured to send path quality notifications to the head-end node once, continually, periodically, in response to configurable changes in path quality, etc. Also, as in the case |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | where multiple primary tunnels are Fast Rerouted, the path quality notification may include an indication of which Fast Rerouted primary tunnels in particular have been effected by the changed path quality." Vasseur '879, 6:7-23.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message (PathErr), the head-end node may wait for the path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. Notably, in the event the head-end node does not receive a path quality notification for the backup tunnel (e.g., within a configurable time limit), the head-end node may attempt to reestablish the new primary tunnel. Moreover, where the head-end node has multiple primary tunnels being Fast Rerouted, a configurable subset of the primary tunnels may be reestablished, e.g., to reduce congestion of the backup tunnels, and/or to limit the number of reestablished primary tunnels within a given period of time." Vasseur '879, 6:24-43.<br><br>"Advantageously, the novel technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. By providing the head-end node of the primary tunnel with path quality feedback of the backup tunnel, the novel technique avoids reestablishing a potentially large number of tunnels over one or more alternate paths after a failure (and Fast Reroute) if the backup tunnels have acceptable path quality. In particular, the backup tunnels, e.g., 0-BW backup tunnels, may not be congested or subsequently burdened by the Fast Rerouted traffic of the primary tunnel. Also, the failed network element (thus the primary tunnel) may be quickly restored; therefore by not reestablishing the primary tunnel, network jitter, churn, etc., may be avoided. Further, the dynamic nature of the novel technique alleviates the need for cumbersome manual configuration." Vasseur '879, 6:44-59. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "FIG. 3 is schematic block diagram of an exemplary signaling (RSVP) message that may be advantageously used with the present invention." Vasseur '879, 7:6-8.<br><br>"FIG. 5 is a schematic block diagram of the computer network in FIG. 1 showing Fast Reroute protection of a primary tunnel using a backup tunnel in accordance with the present invention." Vasseur '879, 7:12-15.<br><br>"FIG. 6 is a schematic block diagram of the computer network in FIG. 5 showing an unacceptable backup tunnel path quality and resultant reestablishing of the primary tunnel in accordance with the present invention." Vasseur '879, 7:16-19.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention." Vasseur '879, 7:20-24.<br><br>"FIG. 1 is a schematic block diagram of an exemplary computer network 100 comprising a plurality of nodes A-F, such as routers or other network devices, interconnected as shown. The nodes may be a part of one or more autonomous systems, routing domains, or other networks or subnetworks. For instance, routers A and E may be provider edge (PE) devices of a provider network, (e.g., a service provider network) that are interconnected to one or more customer networks through customer edge (CE) devices (not shown, while the remaining nodes B-D and F may be core provider (P) devices, as will be understood by those skilled in the art. Those skilled in the art will also understand that the nodes A-F may be any nodes within any arrangement of computer networks, and that the view shown herein is merely an example. For example, the nodes may be configured as connections to/from one or more virtual private networks (VPNs), as will be understood by those skilled in the art. These examples are merely representative. Those skilled in the art will understand that any number of routers, nodes, links, etc. may be used in the computer network 100 and connected in a variety of ways, and that the view shown herein is for simplicity." Vasseur '879, 7:29-49. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Data packets may be exchanged among the computer network 100 using predefined network communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Asynchronous Transfer Mode (ATM) protocol, Frame Relay protocol, Internet Packet Exchange (IPX) protocol, etc. Routing information may be distributed among the routers of the computer network using predetermined Interior Gateway Protocols (IGPs), such as conventional distance-vector protocols or, illustratively, link-state protocols, through the use of IGP Advertisements." Vasseur '879, 7:50-60.<br><br>"FIG. 2 is a schematic block diagram of an exemplary router 200 that may be advantageously used with the present invention, e.g., as an edge router or a core router. The router comprises a plurality of network interfaces 210, a processor 220, and a memory 240 interconnected by a system bus 250. The network interfaces 210 contain the mechanical, electrical and signaling circuitry for communicating data over physical links coupled to the network 100. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols, including, inter alia, TCP/IP, UDP, ATM, synchronous optical networks (SONET), wireless protocols, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), etc." Vasseur '879, 7:61-8:6.<br><br>"The memory 240 comprises a plurality of storage locations that are addressable by the processor 220 and the network interfaces 210 for storing software programs and data structures associated with the present invention. The processor 220 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures. A router operating system 242 (e.g., the Internetworking Operating System, or IOS™, of Cisco Systems, Inc.), portions of which is typically resident in memory 240 and executed by the processor, functionally organizes the router by, inter alia, invoking network operations in support of software processes and/or services executing on the router. These software processes and/or services may comprise routing services 247, Traffic Engineering (TE) services 244, and RSVP services 249. It will be apparent to those skilled in the art that other processor and memory means, including various computer-readable media, may be used to store and execute program instructions pertaining to the inventive technique described herein." Vasseur '879, 8:7-26. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Routing services 247 contain computer executable instructions executed by processor 220 to perform functions provided by one or more routing protocols, such as IGP (e.g., OSPF and IS-IS), IP, BGP, etc. These functions may be configured to manage a forwarding information database (not shown) containing, e.g., data used to make forwarding decisions. Routing services 247 may also perform functions related to virtual routing protocols, such as maintaining VRF instances (not shown) as will be understood by those skilled in the art." Vasseur '879, 8:27-36.<br><br>"RSVP services 249 contain computer executable instructions for implementing RSVP and processing RSVP messages in accordance with the present invention. RSVP is described in RFC 2205, entitled Resource ReSerVation Protocol (RSVP), and in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels, both as incorporated above." Vasseur '879, 8:37-42.<br><br>"TE services 244 contain computer executable instructions for operating TE functions in accordance with the present invention. Examples of Traffic Engineering are described in RFC 3209, RFC 3784, and RFC 3630 as incorporated above, and in RFC 3473, entitled, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions dated January 2003, which is hereby incorporated by reference in its entirety. A TE database (TED, not shown) may be illustratively resident in memory 240 and used to store TE information provided by the routing protocols, such as IGP, BGP, and/or RSVP (with TE extensions, e.g., as described herein), including, inter alia, path quality information as described herein. The TED may be illustratively maintained and managed by TE services 244." Vasseur '879, 8:43-57.<br><br>"Changes in the network topology may be communicated among routers 200 using a link-state protocol, such as the conventional OSPF and IS-IS protocols. Suppose, for example, that a communication link fails or a cost value associated with a network node changes. Once the change in the network's state is detected by one of the routers, that router may flood an IGP Advertisement communicating the change to the other routers in the network. In this manner, each of the routers eventually "converges" to an identical view of the network topology." Vasseur '879, 8:58-67. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "In one embodiment, the routers described herein are IP routers that implement Multi-Protocol Label Switching (MPLS) and operate as label switched routers (LSRs). In one simple MPLS scenario, at an ingress to a network, a label is assigned to each incoming packet based on its forwarding equivalence class before forwarding the packet to a next-hop router. At each router, a forwarding selection and a new substitute label are determined by using the label found in the incoming packet as a reference to a label forwarding table that includes this information. At the network egress, a forwarding decision is made based on the incoming label but optionally no label is included when the packet is sent on to the next hop. In some network configurations, one hop prior to the network egress, a penultimate hop popping (PHP) operation may be performed. Particularly, because the hop prior to the network egress (the penultimate hop) is attached to the network egress, the label is no longer needed to assure that the traffic follows a particular path to the network egress. As such, the PHP-enabled device "pops" the labels from the traffic before forwarding the traffic to the network egress, e.g., using conventional or native (IP) routing, thereby alleviating the task of removing the labels at the network egress." Vasseur '879, 9:1-23.<br><br>"The paths taken by packets that traverse the network in this manner are referred to as label switched paths (LSPs) or Traffic Engineering (TE)-LSPs. An example TE-LSP is shown as the thick line and arrow (T1) between a head-end node (router A) and a tailend node (router E) in FIG. 1. Establishment of a TE-LSP requires computation of a path, signaling along the path, and modification of forwarding tables along the path. MPLS TE establishes LSPs that have guaranteed bandwidth under certain conditions. Illustratively, the TE-LSPs may be signaled through the use of the RSVP protocol (with Traffic Engineering extensions), and in particular, RSVP TE signaling messages. Notably, when incorporating the use of PCEs (described below), the path computation request (and response) between PCC and PCE can be exchanged in accordance with a protocol specified in Vasseur, et al., Path Computation Element (PCE) Communication Protocol (PCEP)—Version 1—<draft-vasseur-pce-pcep-02.txt>, Internet Draft, September 2005, the contents of which are hereby incorporated by reference in its entirety. It should be understood that the use of RSVP or PCEP serves only as an example, and that other communication protocols may be used in accordance with the present invention." Vasseur '879, 9:24-45. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "In accordance with RSVP, to request a data flow (TE-LSP) between a sender and a receiver, the sender may send an RSVP path request (Path) message downstream to the receiver along a path (e.g., a unicast route) to identify the sender and indicate e.g., bandwidth needed to accommodate the data flow, along with other attributes of the TE-LSP. The Path message may contain various information about the data flow including, e.g., traffic characteristics of the data flow. Also in accordance with the RSVP, a receiver establishes the TE-LSP between the sender and receiver by responding to the sender's Path message with a reservation request (Resv) message. The reservation request message travels upstream hop-by-hop along the flow from the receiver to the sender. The reservation request message contains information that is used by intermediate nodes along the flow to reserve resources for the data flow between the sender and the receiver, to confirm the attributes of the TE-LSP, and provide a TE-LSP label. If an intermediate node in the path between the sender and receiver acquires a Path message or Resv message for a new or established reservation (TE-LSP) and encounters an error (e.g., insufficient resources, failed network element, etc.), the intermediate node generates and forwards a path or reservation error (PathErr or ResvErr, hereinafter Error) message to the sender or receiver, respectively." Vasseur '879, 9:46-10:2.<br><br>"FIG. 3 is a schematic block diagram of portions of a signaling message 300 (e.g., RSVP message, such as Path, Resv or Error) that may be advantageously used with the present invention. Message 300 contains, inter alia, a common header 310 and one or more signaling protocol specific objects 320, such as an LSP-ATTRIBUTE object 330. The common header 310 may comprise a source address 312 and destination address 314, denoting the origination and requested termination of the message 300. Protocol specific objects 320 contain objects necessary for each type of message 300 (e.g., Path, Resv, Error, etc.). For instance, a Path message may have a sender template object, Tspec object, Previous-hop object, etc. The LSP-ATTRIBUTE object 330, for instance, may be used to signal attributes and/or information regarding an LSP (tunnel). To communicate this information, LSP-ATTRIBUTE object 330 (as well as specific objects 320) may include various type/length/value (TLV) encoding formats and/or flags, as will be understood by those skilled in the art. An example of an LSP-ATTRIBUTE object is further described in Farrel, et al., Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE <draft-ietf-mpls-rsvpte-attributes-05.txt>, |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | Internet Draft, May 2005, which is hereby incorporated by reference as though fully set forth herein. A Resv message, on the other hand, may have specific objects 320 for a label object, session object, filter spec object, etc., in addition to the LSP-ATTRIBUTE object 330. Error messages 300 (e.g., PathErr or ResvErr), may also have specific objects 320, such as for defining the type of error, etc." Vasseur '879, 10:3-31.<br><br>"It should be noted that in accordance with RSVP signaling, the state of the TE-LSP is refreshed on a timed interval, e.g., every thirty seconds, in which RSVP Path and Resv messages are exchanged. This timed interval is configurable by a system administrator. Moreover, various methods understood by those skilled in the art may be utilized to protect against route record objects (RROs) contained in signaling messages for a TE-LSP in the event security/privacy is desired. Such RRO filtering prevents a head-end node of the TE-LSP from learning of the nodes along the TE-LSP, i.e., nodes within the provider network." Vasseur '879, 10:4-42.<br><br>"Although the illustrative embodiment described herein is directed to MPLS, it should also be noted that the present invention may advantageously apply to Generalized MPLS (GMPLS), which pertains not only to packet and cell-based networks, but also to Time Division Multiplexed (TDM) and optical networks. GMPLS is well known and described in RFC 3945, entitled Generalized Multi-Protocol Label Switching (GMPLS) Architecture, dated October 2004, and RFC 3946, entitled Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control, dated October 2004, the contents of both of which are hereby incorporated by reference in their entirety." Vasseur '879, 10:43-55.<br><br>"To obviate delays associated with updating routing tables when attempting to avoid a failed network element (i.e., during convergence), some networks have employed MPLS TE Fast Reroute (FRR). MPLS Fast Reroute is a technique that may be used to quickly divert ("Fast Reroute") traffic around failed network elements in a TE-LSP. MPLS Fast Reroute is further described, for example, by Fast Reroute Extensions to RSVP-TE for LSP Tunnels, as incorporated by reference above. According to the technique, one or more network elements (e.g. links or nodes) in a network are protected by backup tunnels following an alternate |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | path. If a failure occurs on a protected link or node, TE-LSPs (and consequently the traffic that they carry) are locally diverted onto an appropriate alternate path (e.g., a "backup tunnel") by the node immediately upstream from the failure. The backup tunnel acts as a Fast Reroute path for the primary TE-LSP and obviates delays associated with other measures, such as tearing down the primary TE-LSP after having gracefully diverted the TE-LSPs affected by the failure, should an alternate path around the failed network element exist. In the event of a failure of a protected element the head-end node of the backup tunnel (or a "point of local repair," PLR node) may quickly begin diverting traffic over the backup tunnel with minimal disruption to traffic flow. Those skilled in the art will understand that MPLS Fast Reroute is one example of link or node failure protection, and that other known correction mechanisms may be used in accordance with the present invention. As mentioned above, however, the head-end node of the Fast Rerouted primary tunnel may attempt to reestablish the primary tunnel in response to learning of the protected element failure, particularly in the case where the backup tunnel is a zero-bandwidth (0-BW) tunnel. The attempt to reestablish the primary tunnel has conventionally been a systematic response to Fast Rerouting (diverting) of the primary tunnel, regardless of the path quality of the backup tunnel." Vasseur '879, 10:56-11:23.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 11:24-39. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as RSVP TE signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or TLV encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object 330 of the RSVP messages 300 to convey the path quality notification." Vasseur '879, 11:40-49.<br><br>"FIG. 4 is a schematic block diagram illustrating the format of an extension object (TLV) 400, such as a Fast Reroute Feedback object, that may be advantageously used with the present invention. The extension object (or sub-object) 400 is illustratively embodied as a TLV contained in an LSP-ATTRIBUTE object 330 of an RSVP message 300 and is extended to carry backup (and/or primary) tunnel path quality information. To that end, the "Feedback" object 400 is organized to include a Type field 405 containing a predetermined type value signifying the specific content of the object 400. The Length field 410 is a variable length value. The TLV encoded format may also comprise one or more non-ordered sub-TLVs 450 carried within the TLV "payload" (e.g. Value field 415), each having a Type field 455, Length field 460, and Value field 465. The fields of the TLV 400 and sub-TLV(s) 450 are used in a variety of manners, including as described herein, according to the present invention." Vasseur '879, 11:59-12:8.<br><br>"In accordance with one aspect of the present invention, the head-end node (e.g., router A) requests the establishment of the primary tunnel (e.g., T1), such as a TE-LSP. Notably, the head-end node may be a head-end node for multiple primary tunnels, as will be understood by those skilled in the art. Along with the primary tunnel establishment, the head-end node may also request Fast Reroute protection of one or more network elements (e.g., all intermediate network elements) at a PLR node (e.g., router B as shown). Note that each intermediate node along the primary tunnel may act as a PLR node, and that router B is shown merely for simplicity. Illustratively, the Fast Reroute protection may be embodied as one or more zero-bandwidth (0-BW) backup tunnels at the PLR node (e.g., BT1). Those skilled in the art will also understand that the PLR node may protect more than one primary |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | tunnel originating at more than one corresponding head-end node (not shown)." Vasseur '879, 12:9-25.<br><br>"The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. For example, various connectivity verification protocols, such as, e.g., Bidirectional Forwarding Detection (BFD), IGP "Hello" packets, BGP KEEPALIVE messages, etc., may be used to detect a failure of a network element, as will be understood by those skilled in the art. Furthermore, other lower-layer failure detection mechanisms (e.g. optical or SONET/SDH alarms) may be used to detect a network element failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and may send an error message (e.g., an RSVP PathErr 300, such as a conventional "tunnel locally repaired" message) to the head end node indicating the "Fast Rerouting" of the primary tunnel. FIG. 5 is a schematic block diagram of the computer network 100 in FIG. 1 showing Fast Reroute protection of the primary tunnel T1 (e.g., in response to a protected network element, router C, failure, indicated with an overlaid "X") using a backup tunnel BT1 in accordance with the present invention. Traffic originally received at the PLR node (router B) over the primary tunnel is now diverted over the backup tunnel to a remerge point (router D) of the primary tunnel, as will be understood by those skilled in the art." Vasseur '879, 12:42-65.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, each PLR node may collect metrics/statistics of the primary and/or backup tunnels. For instance, example metrics may comprise, inter alia, packet drops, path cost, jitter, delay, bandwidth, etc. The PLR node may collect the metrics through traffic monitoring, probes, independent calculations, and/or through cooperation with protected nodes of the primary tunnel (primary nodes) and nodes of the backup tunnel (backup nodes), e.g., transmitting path quality notifications. Once the primary tunnel is Fast Rerouted (i.e., diverted after failure of a protected network element), the PLR node continues to collect metrics of the backup tunnel. (Alternatively, metrics of the backup tunnel may be collected only after Fast Reroute, and not prior to Fast Reroute.)" Vasseur '879, 12:66-13:12. |

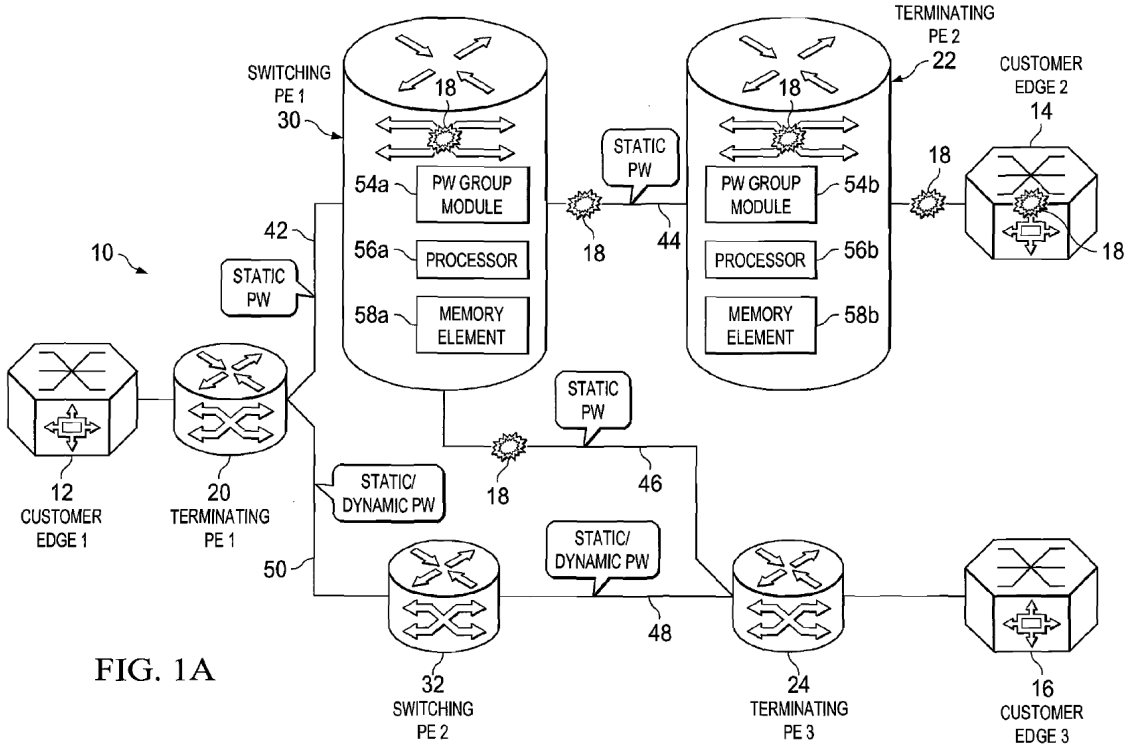
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "For example, a path cost increase from the primary tunnel to the backup tunnel may be computed (and transmitted, below) by the PLR node prior to the failure (or during the failure while the primary TE-LSP is diverted onto the backup tunnel) using its own routing tables. The path cost increase may be calculated as a difference between the entire length (head-end node to tail-end node) of the primary and backup tunnels, or just the difference between the protected segment of the primary tunnel and the backup tunnel (PLR node to remerge point). Also, a jitter increase of the primary and backup tunnels, which may be generally described as a difference between inter-arrival of packets, may be monitored using various known techniques, such as, e.g., sending probe packets (probes) from the PLR node to the remerge point. For instance, probes may determine that packets arrive at the remerge point along the primary tunnel from the PLR node consistently, e.g., every 10 milliseconds (ms) (e.g., an average value). After Fast Reroute, however, probes may determine that packets do not arrive at the remerge point along the backup tunnel from the PLR node consistently, e.g., one may arrive in 10 ms, another in 100 ms, another in 50 ms, etc. The non-constant rate of received packets (jitter) may be undesirable, e.g., in particular for voice over IP (VoIP) traffic, as will be understood by those skilled in the art." Vasseur '879, 13:13-36.<br><br>"As a further example, packet dropping may be monitored for the primary and backup tunnels prior to and after Fast Reroute (respectively). For instance, based on the tunnel label of the dropped packet, primary nodes and/or backup nodes may be able to distinguish which tunnel corresponds to the dropped packets. Each of the primary and/or backup nodes collect packet drop data, and periodically inform the PLR node of the number of dropped packets (e.g., though a corresponding Feedback object 400). In the case of a backup node, the PLR node receiving the notification may interpret the association of the backup tunnel label and the primary tunnel label to reference an appropriate primary tunnel. Those skilled in the art will understand that the above path quality metrics are merely examples, and that any other metrics/statistics useful for determining path quality of the backup tunnel may be used in accordance with the present invention (e.g., delay, bandwidth, etc.). Further, the path quality information may be measured and compared in a variety of manners, such as, e.g., as a difference between primary and backup tunnels before and after Fast Reroute, or simply the difference between the backup tunnel before and after Fast Reroute, etc." Vasseur '879, 13:37-58. |

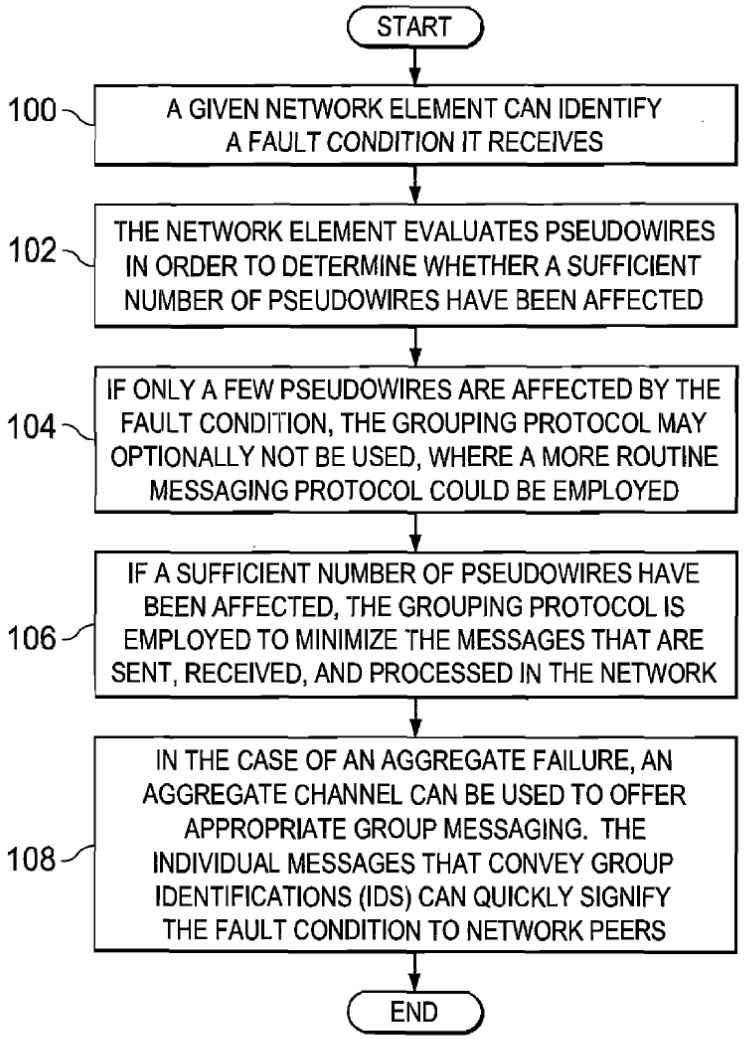
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "Also after the primary tunnel is Fast Rerouted, the PLR node may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. For instance, the novel Feedback object 400 may include one or more sub-TLVs 450 corresponding to metrics/statistics, as described above. Notably, the path quality information pertaining to a particular metric/statistic may be transmitted as total values for interpretation by the head-end node (e.g., to determine the difference), or as PLR-node-computed differences (e.g., between the primary and backup tunnels, or before and after Fast Reroute). For instance, if the delay of the primary tunnel (along the protected segment) prior to Fast Reroute were 2 ms, and after Fast Reroute the delay of the backup tunnel were 5 ms, the notification may be configured to include both values 2 ms and 5 ms, or instead simply the difference, i.e., an increase of 3 ms." Vasseur '879, 13:59-14:8.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message 300 (PathErr), the head-end node may wait for at least one path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. Also, any number of metrics may be used in the determination, e.g., as configured by a system administrator. For example, using the metrics described above, a head-end node may be configured to reestablish the primary tunnel in response to i) a certain number of packet drops, ii) a percent increase in packet drops, iii) a number of packet drops and a percent increase in path cost, iv) a percent increase in path cost and a percent increase in jitter, etc. Those skilled in the art will understand that these are merely examples of possible path quality comparisons and reestablishment determinations, and that any comparisons to any metrics at any configurable changes may be used in accordance with the present invention." Vasseur '879, 14:41-59.<br><br>"If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. FIG. 6 is a schematic block diagram of the computer network 100 in FIG. 5 showing an unacceptable backup tunnel path quality (dotted line and arrow) and resultant reestablishment of the primary tunnel in |

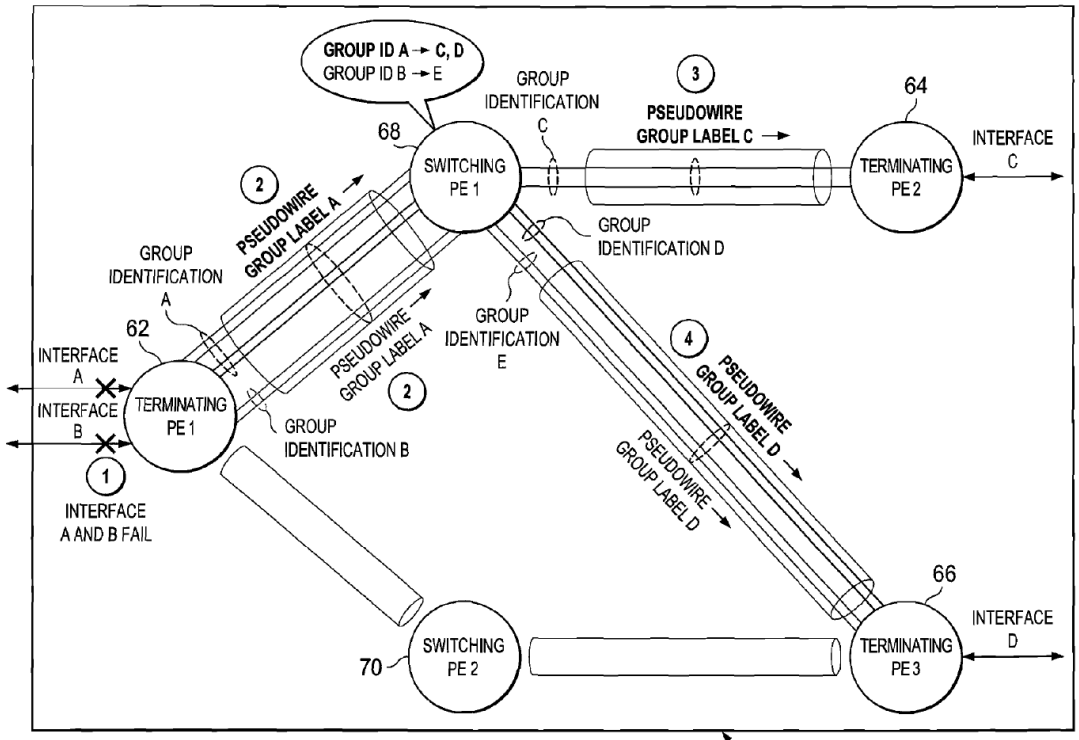
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | accordance with the present invention. Those skilled in the art will understand that the reestablished primary tunnel may traverse one or more primary nodes (not shown), and that it may be computed specifically to avoid the failed network element and any network elements of the unacceptable backup tunnel. Those skilled in the art will also understand that the attempt to reestablish the primary tunnel may not be able to find an acceptable path, in which case the head-end node may continue to use the unacceptable backup tunnel or other unacceptable rerouted path." Vasseur '879, 14:60-15:9.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention. The procedure 700 starts at step 705, and continues to step 710, where a head-end node (e.g., router A) establishes a primary tunnel (e.g., T1) to a destination tail-end node (e.g., router E). During or after establishment, the head-end node requests protection of the primary tunnel at step 715, and in response, PLR nodes along the primary tunnel (e.g., router B) create backup tunnels (e.g., BT1) to protect the primary tunnel in step 720. (Those skilled in the art will understand that backup tunnels around the protected network elements may already exist at the PLR node, and that "creating backup tunnels" in step 720 implies an association with pre-existing backup tunnels.) As mentioned above, these backup tunnels may illustratively be embodied as 0-BW backup tunnels. In accordance with the present invention, the head-end node may additionally request backup tunnel path quality notification from the PLR nodes in step 725, such as, e.g., through the use of empty corresponding Feedback objects in RSVP (Path) messages 300, as described above." Vasseur '879, 15:37-58.<br><br>"The procedure 700 continues to FIG. 7B (step "A"), where in step 750 the PLR node detecting the failure diverts ("Fast Reroutes") the primary tunnel traffic to the backup tunnel and sends an error message (e.g., an RSVP (Error) message 300) to the head end node, e.g., a "tunnel locally repaired" message. The detecting PLR node continues to collect path quality information for the backup tunnel in step 755 and at step 760 determines whether to send the path quality notification to the head-end node. For example, as mentioned above, the PLR node may be configured to continually send notifications, or periodically, or in response to a configurable change in path quality, etc. Also as mentioned above, the PLR |

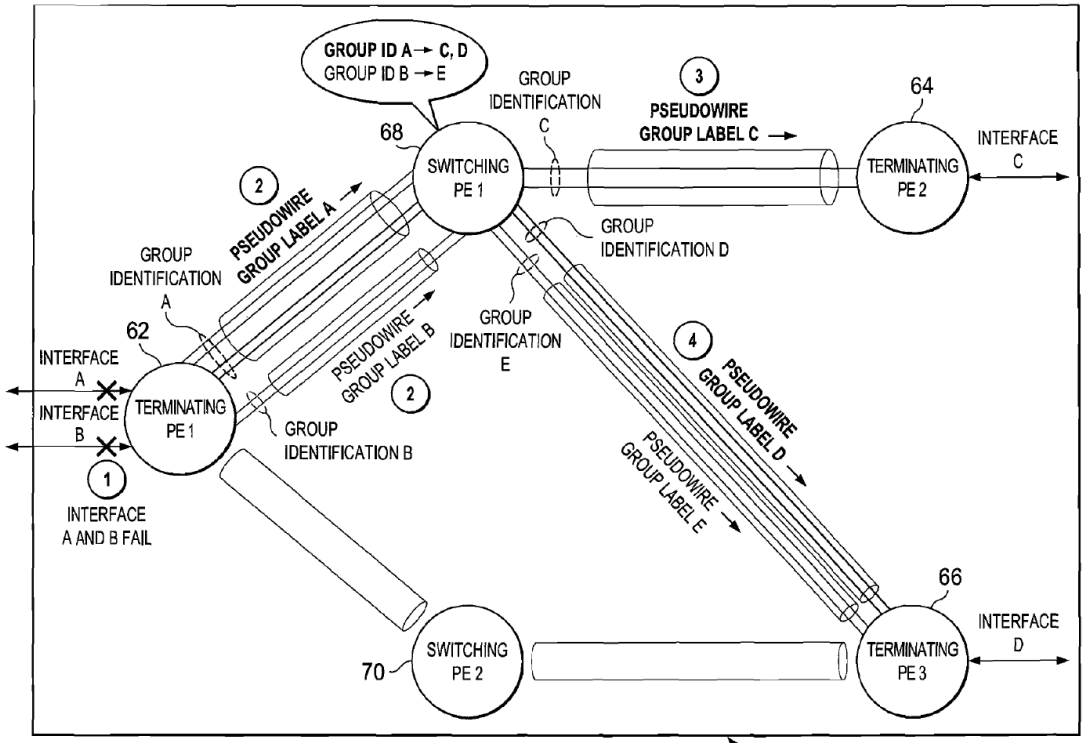
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | node may be configured to send either the actual path quality information or the change (difference) in path quality. (As further mentioned above, metrics of the backup tunnel may alternatively be collected only after Fast Reroute, and not prior to Fast Reroute.)." Vasseur '879, 16:4-20.<br><br>"If the PLR node decides to send the notification in step 760, then the head-end node determines whether to reestablish the primary tunnel based on the backup tunnel path quality notification in step 765, e.g., based on one or more configurable thresholds, percentages, etc., as described above. If the backup tunnel is currently maintaining an acceptable quality for the traffic flow in step 770, the head-end node may continue to utilize the backup tunnel, and the PLR node continues to collect path quality information in step 755 to detect any change in quality. Otherwise, if the backup tunnel quality is not acceptable in step 770, the head-end node may attempt to reestablish the primary tunnel in step 775. Notably, as mentioned above, if the PLR node has not sent any notification (step 760) within a configurable period of time in step 780, e.g., due to a backup tunnel failure, over-congestion, etc., then the head-end node may also attempt to reestablish the primary tunnel in step 775 accordingly. Moreover, as described above, in the event more than one primary tunnel is Fast Rerouted for the head-end node, various techniques to reestablish one or more of the primary tunnels may be used (e.g., as many tunnels as necessary, a configurable subset of tunnels, all tunnels, the congested tunnels, etc.). The procedure 700 ends in step 785." Vasseur '879, 16:21-43.<br><br>"While there has been shown and described an illustrative embodiment that dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the present invention. For example, the invention has been shown and described herein using "Fast Reroute," e.g., MPLS TE Fast Reroute (FRR). However, the invention in its broader sense is not so limited, and may, in fact, be used with other network element protection and failure correction mechanisms as will be understood by those skilled in the art. Moreover, while the above description describes performing the technique at the head-end node and PLR node, the invention may also be advantageously used with PCEs. In addition, while one |

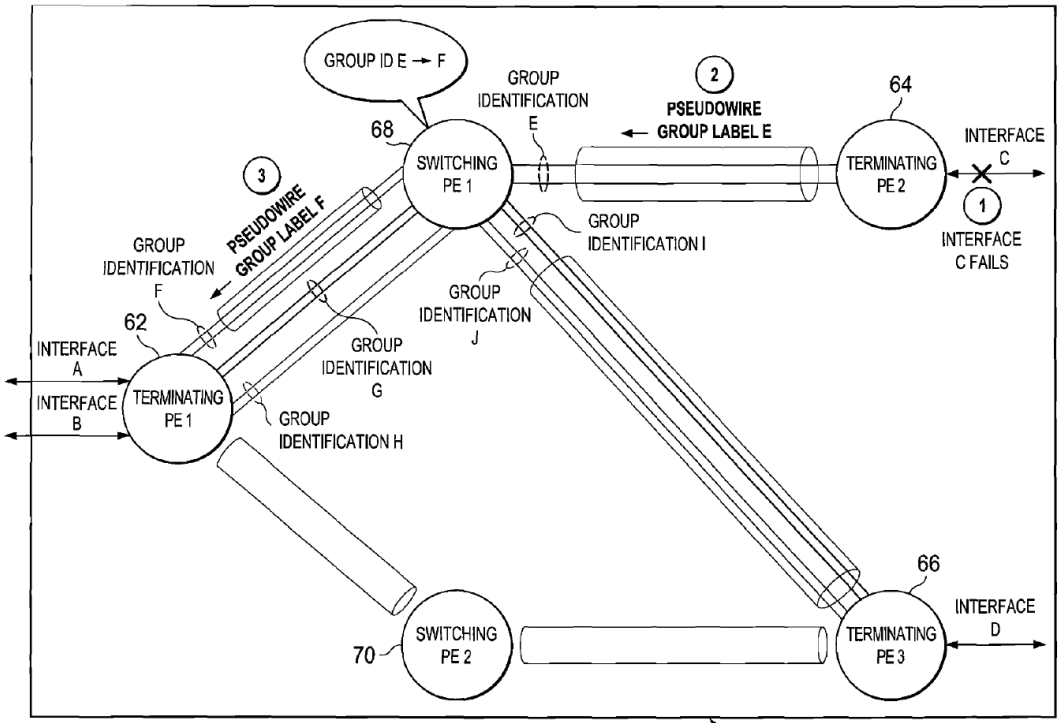
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 16:63-17:13.<br><br>**Rustogi discloses**:<br>"An example method includes identifying a fault condition in a network, and evaluating pseudowires affected by the fault condition in order to make a determination as to whether an aggregate failure occurred in the network for a group of pseudowires. The method also includes communicating a group message indicating that the group of pseudowires is associated with the fault condition. The group message includes a group identification (ID), which identifies the group of pseudowires, and the group message includes a pseudowire group label identifying an in-band aggregate channel. More specifically, the pseudowire group label can be applicable to static pseudowires. In more detailed embodiments, the group ID identifies the group of pseudowires that are associated with an attachment circuit, a label switched path, or a port. Internal mappings can be maintained such that a plurality of pseudowires is mapped to individual interfaces of network elements in the network." Rustogi, Abstract. |

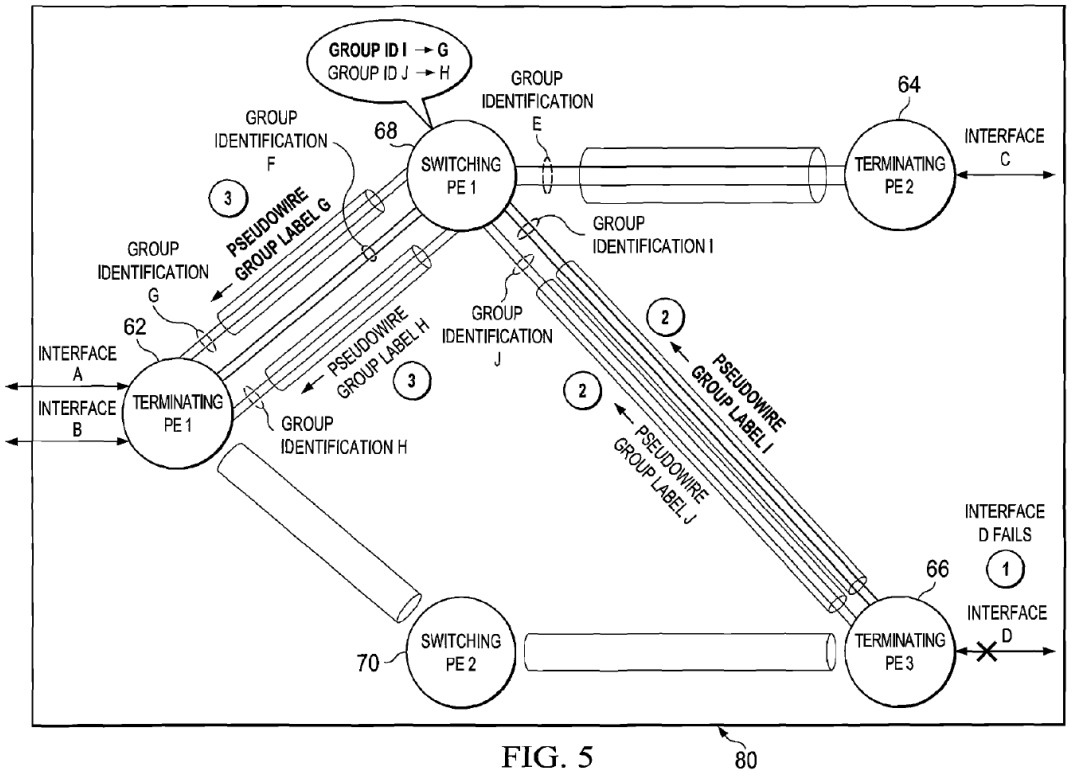
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 1A<br>Rustogi, FIG. 1A. |

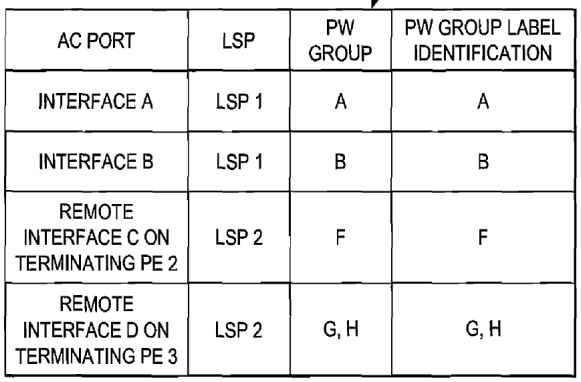
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 1B<br>Rustogi, FIG. 1B. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Rustogi, FIG. 2. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Rustogi, FIG. 3. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Rustogi, FIG. 4. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | |  <br>FIG. 5 80<br>Rustogi, FIG. 5. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | 74<br>  <br>FIG. 6<br>Rustogi, FIG. 6.<br><br>"The field of communications has become increasingly important in today's society. In particular, the ability to quickly and to effectively provision connections presents a significant challenge to component manufacturers, system designers, and network operators. Multiprotocol Label Switching (MPLS) is a mechanism in telecommunications networks that carries data from one network node to the next. Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be emulated over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDUs) and transmitting them over pseudowires. Protocols exist for establishing and maintaining the pseudowires. Certain issues have arisen in pseudowire scenarios, where faults are detected in the network." Rustogi, ¶ [0002].<br><br>"FIG. 1A is a simplified block diagram of a communication system for providing pseudowire group labels in a network environment in accordance with one embodiment of the present disclosure." Rustogi, ¶ [0004]. |

| AC PORT | LSP | PW GROUP | PW GROUP LABEL IDENTIFICATION |
|---|---|---|---|
| INTERFACE A | LSP 1 | A | A |
| INTERFACE B | LSP 1 | B | B |
| REMOTE INTERFACE C ON TERMINATING PE 2 | LSP 2 | F | F |
| REMOTE INTERFACE D ON TERMINATING PE 3 | LSP 2 | G, H | G, H |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "FIG. 1B is a simplified flowchart depicting one possible, generic operational flow associated with the communication system." Rustogi, ¶ [0005].<br><br>"FIG. 2 is a simplified block diagram of an example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0006].<br><br>"FIG. 3 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0007].<br><br>"FIG. 4 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0008].<br><br>"FIG. 5 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0009].<br><br>"FIG. 6 is a simplified table of an example set of pseudowire group provisioning parameters in accordance with one embodiment." Rustogi, ¶ [0010].<br><br>"FIG. 1A is a simplified block diagram of a communication system 10 for providing pseudowire group labels in accordance with one example of the present disclosure. FIG. 1A includes a customer edge 1 (CE1) 12, a CE2 14, and a CE3 16, where a number of faults 18 are shown as propagating in the network. Typically, when an error or a failure occurs in the network (e.g., an interface failure, a pulled cable, a switch failure, hardware/software failures generally, etc.), messages are sent to various network devices in order to inform them of these fault conditions. Faults 18 of FIG. 1A are indicative of such messages, where the underlying fault condition (being signaled by the messages) can occur virtually anywhere in a network (e.g., in a customer edge, in provider equipment, etc.). FIG. 1A also includes a terminating provider equipment 1 (TPE1) 20, a TPE2 22, a TPE3 24, a switching provider edge 1 (SPE1) 30, and a SPE2 32. In one particular example implementation, the TPEs and SPEs of FIG. 1A are switches that are configured to exchange data in a network environment." Rustogi, ¶ [0012]. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "SPE1 30 may include a pseudowire (PW) group module 54 a, a processor 56 a, and a memory element 58 a. In a similar fashion, TPE2 22 may include a pseudowire group module 54 b, a processor 56 b, and a memory element 58 b. FIG. 1A also includes a number of static pseudowires 42, 44, and 46. A set of static/dynamic pseudowires 48, 50 is also provided. Note that the group labeling protocol discussed herein can be executed between individual SPEs, TPEs, or between any combinations of these elements." Rustogi, ¶ [0013].<br><br>"In one particular arrangement, communication system 10 is provided in conjunction with a Layer-2 virtual private networks (L2VPN)/operation, administration, and maintenance (OAM) L2VPN/OAM framework. The OAM framework is intended to provide OAM layering across L2VPN services, pseudowires, and packet switched network (PSN) tunnels. Communication system 10 may include any suitable networking protocol or arrangement that provides a communicative platform for communication system 10. Thus, communication system 10 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Communication system 10 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs." Rustogi, ¶ [0014].<br><br>"Failure detection and failure notification for static pseudowires is inadequate, where sluggish signaling can result in poor scalability for failover performance. Typically, static pseudowires are manually configured at respective endpoints, where control channels are absent for providing group level signaling messages. Aggregate channels are significant tools for providing suitable scalability in the network, but no such aggregate channel exists for static pseudowires. For dynamic pseudowires, such an aggregate channel may be present in the form of a label distribution protocol (LDP) directed session. However, no such protocol exists for static pseudowire configurations such that an in-band aggregate channel would be available for static pseudowires." Rustogi, ¶ [0016].<br><br>"Communication system 10 can address the aforementioned issues (and others) by offering a pseudowire group label that can represent an aggregate channel for groups of static pseudowires. The aggregate channel of communication system 10 can allow for improved |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | scalability of failover performance. In accordance with one potential configuration of communication system 10, a pseudowire group label is representative of a group of static pseudowires transported over a label switched path (LSP). The pseudowire group label can identify the aggregate channel, which captures the hierarchy relevant to OAM mechanisms. Additionally, the groups represented by the group identification (ID) can be mutually exclusive, where a pseudowire is part of only one group. In other embodiments, a pseudowire can be part of multiple groups, or be configured in any other suitable manner based on particular network arrangements." Rustogi, ¶ [0017].<br><br>"During operations, and with brief reference to FIG. 1B, a given network element can identify a fault condition it receives (at step 100) and, subsequently, evaluate pseudowires in order to determine whether a sufficient number of pseudowires have been affected. This is reflected by step 102. If only a few pseudowires are affected by the fault condition, the grouping protocol outlined herein may have only nominal value, where there could be an option to simply communicate the fault condition in a more routine manner, as outlined in step 104. However, if a sufficient number of pseudowires have been affected, the grouping protocol outlined herein can be employed to minimize the messages that are sent, received, and processed in the network. This is reflected as step 106. Note that the determination (as to whether a sufficient number of pseudowires have been impacted by the fault condition) can involve accessing internal tables such that a quick mapping can occur to determine if an aggregate failure has occurred. As used herein, the term 'aggregate failure' simply connotes that a sufficient number of pseudowires have experienced the fault condition such that an aggregate channel can be employed to offer appropriate group messaging. For the aggregate failure condition, the individual messages that convey Group identifications (IDs) can quickly signify the fault condition to network peers, as shown in step 108." Rustogi, ¶ [0018].<br><br>"In specific regards to OAM mechanisms, OAM messages typically result from common failures in the network. These fault conditions can be aggregated such that they are signaled as a single message, which could represent a group of failed pseudowires (as opposed to sending individual messages for each failed pseudowire). Hence, a single message could be sent to represent all the relevant OAM messages propagating in communication system 10. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | The group label that propagates in communication system 10 provides an architecture with a significant level of aggregation for failed pseudowires (i.e., pseudowires being affected by a given fault condition), particularly for OAM messaging. Moreover, the in-band aggregate channel of communication system 10 is based (at least in part) on the evolving trends of OAM mechanisms, which are required to be fast, responsive, and capable of being implemented in hardware or software. Additionally, in-band OAM protocols are a better measure of the path availability." Rustogi, ¶ [0020].<br><br>"In operation of one example implementation, a group label can represent the tuple <attachment circuit (AC) port level grouping, LSP>. This could signify that all pseudowires on an AC port (sought for aggregation) traverse a given LSP. Multiple pseudowire groups can exist within an LSP. Similarly, pseudowires on the same AC port (that traverse a different LSP) can use a different pseudowire group label. Alternatively, an administrator may seek to employ a one-to-one mapping between an LSP and a group label. If that were the case, then only one pseudowire group would exist within an LSP. In scenarios where there is no LSP label in the packet (e.g., due to penultimate hop popping), the pseudowire group label can provide the hierarchy that is appropriate." Rustogi, ¶ [0021].<br><br>"In one particular example, the group level pseudowire OAM message can be sent with the following label stack: Explicit/Implicit LSP Label+pseudowire group Label+GAL+ACH+pseudowire OAM with grouping TLV (where GAL=Generic Associated Channel Label, ACH=Associated Channel Header, TLV=Type-Length-Value). If there are multiple LSPs, then one group label can be provisioned for each LSP (for each pseudowire group), where per group messages can be sent on each LSP. The group label does not necessarily have a one-to-one mapping to the grouping of pseudowires implied by the Group ID in the grouping TLV. Note also that the group-based aggregate channel is applicable to static pseudowires, as well as for dynamic pseudowires in certain applications." Rustogi, ¶ [0022].<br><br>"As discussed herein, the aggregate channel of communication system 10 can be configured in various ways. For example, and with regards to a first option, a separate label may simply be used to identify a pseudowire group within an LSP. The association of an OAM message |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | and a pseudowire group is straightforward. There could potentially be multiple pseudowire group labels per LSP. As a second option, one group label can be used to identify a common pseudowire group channel on the LSP. In this implementation, one pseudowire group label is provided per LSP. The OAM message association to a pseudowire group is not as simple as the first option. As a third option, one pseudowire is simply designated to convey grouping information (e.g., without using a group label). In this case, there is no need for a pseudowire group label. Again, the OAM message association to a pseudowire group is not as simple as the first option." Rustogi, ¶ [0023].<br><br>"Any combination of formatting (for the Group ID and the pseudowire group label) can be used in the group message to be communicated in the network. In one example, only one of these elements is communicated when an aggregate fault condition is detected, or these elements can be combined into a single unique identifier. In the most generic example, a group message would at least include the Group ID (identifying the pseudowires affected by the fault) and a pseudowire group label (identifying an aggregate channel for communicating the group message). In this generic sense, a pipe (the Group ID) within a pipe (the pseudowire group label) is being identified, where the group message is identifying both elements during an aggregate fault condition. Operational details of communication system 10 are described below with reference to FIGS. 2-6. Note that before turning to additional example flows and example embodiments of the present disclosure, a brief overview of the infrastructure of communication system 10 is provided." Rustogi, ¶ [0024].<br><br>"CE1 12, CE2 14, and CE3 16 represent devices, infrastructure, equipment, clients, or customers seeking to initiate a data session in communication system 10. These elements may can comprise a digital subscriber line access multiplexer (DSLAM), a router, a personal computer, a server, a switch, and/or other devices associated with data propagation. Further, these elements may sit behind, or in front of, one or more of these identified devices. The term 'CE' may be inclusive of the devices identified above (e.g., a DSLAM, a switch, etc.), as well as devices used to initiate a communication, such as a console, a proprietary endpoint, a telephone, a cellular telephone, a bridge, a computer, a personal digital assistant (PDA), a laptop or an electronic notebook, or any other device, component, element, or object capable of initiating voice, audio, media, or data exchanges within communication |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | system 10. The customer element may also include any device that seeks to initiate a communication on behalf of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating a voice, a video, text, or a data exchange within communication system 10. Data, as used herein in this document, refers to any type of video, numeric, voice, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another." Rustogi, ¶ [0025].<br><br>"SPE1 30, SPE2 32, TPE1 20, TPE2 22, and TPE3 24 are network elements that facilitate communications in two directions in a network environment. In one particular example, each of these network elements is a switch configured to exchange data over static and/or dynamic pseudowire links. Further, the traffic exchanged between these components may be directed over an MPLS transport in certain embodiments. As used herein in this Specification, the term 'network element' is meant to encompass switches, routers, bridges, gateways, servers, processors, loadbalancers, firewalls, or any other suitable device, component, element, or object operable to exchange or process information in a network environment. Moreover, these network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information. Along similar design alternatives, any of the internal modules and components of these network elements may be combined in various possible configurations." Rustogi, ¶ [0029].<br><br>"Turning to FIG. 2, FIG. 2 is a simplified block diagram of an example system 60 for providing an example use case using per-label switched path (LSP) pseudowire group labels. FIG. 2 includes a TPE1 62, a TPE2 64, a TPE3 66, a SPE1 68, and a SPE2 70. Each pseudowire group is identified, where a group identification (ID) for Group A and Group B is depicted at TPE1 62. Similarly, Groups C, D, and E have Group IDs at SPE1 68. TPE2 64 and TPE3 66 can couple to interfaces C and D, respectively." Rustogi, ¶ [0031]. |

| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "In this particular example, interfaces A and B have failed. Note that there is a multitude of attachment circuits (e.g., 1000 attachment circuits) that are being transported over these interfaces A and B, where the attachment circuits are being tunneled into a corresponding number of pseudowires. For example, there could be 500 attachment circuits on interface A (implicating 500 pseudowires) and 500 attachment circuits on interface B, where the fault condition for the pseudowires should be signaled. In other flawed systems, an architecture would individually signal this fault condition for each pseudowire (e.g., via signaling between TPE1 62 and SPE1 68). Instead of sending 500 messages, a single message can be sent, where a single label (and Group ID) can be used to identify the pseudowires. In this case, the Group ID A is used to signal the fault condition for 300 pseudowires and for 200 pseudowires (i.e., the top two links connecting TPE1 62 and SPE1 68) using a single message (that includes Group Label A and Group ID A). Thus, the status for Group A is quickly communicated to SPE1 68. Similarly, Group ID B can be used to signal the status of the other 500 pseudowires to appropriately convey the status for Group B. More specifically, the message can include Group Label A and Group ID B. Note that all 1000 pseudowires have effectively been accounted for using these Group IDs A and B." Rustogi, ¶ [0032].<br><br>"FIG. 3 is a simplified block diagram of an example system 72 for providing another use case for pseudowire group labels. Note that the grouping mechanism outlined herein is not limited to pseudowires that propagate over LSPs. Certain pseudowires can propagate over an LSP and represent one group, where two ports can be provisioned for two different groups (e.g., Group A and Group B). Hence, FIG. 3 is depicting a use case using pseudowire group labels for <port, LSP>mapping. In a general sense, such a configuration is showing how pseudowire mechanics can be used to offer different group signaling, which may be based on various possible implementations. Thus, there is a group level construct corresponding to the group labels that are created such that any OAM protocol can send the appropriate aggregate messages. In this particular example, the signaling for Group ID A, B, C, and D is similar to that of FIG. 2; however, the grouping mechanism has simply changed." Rustogi, ¶ [0035]. |

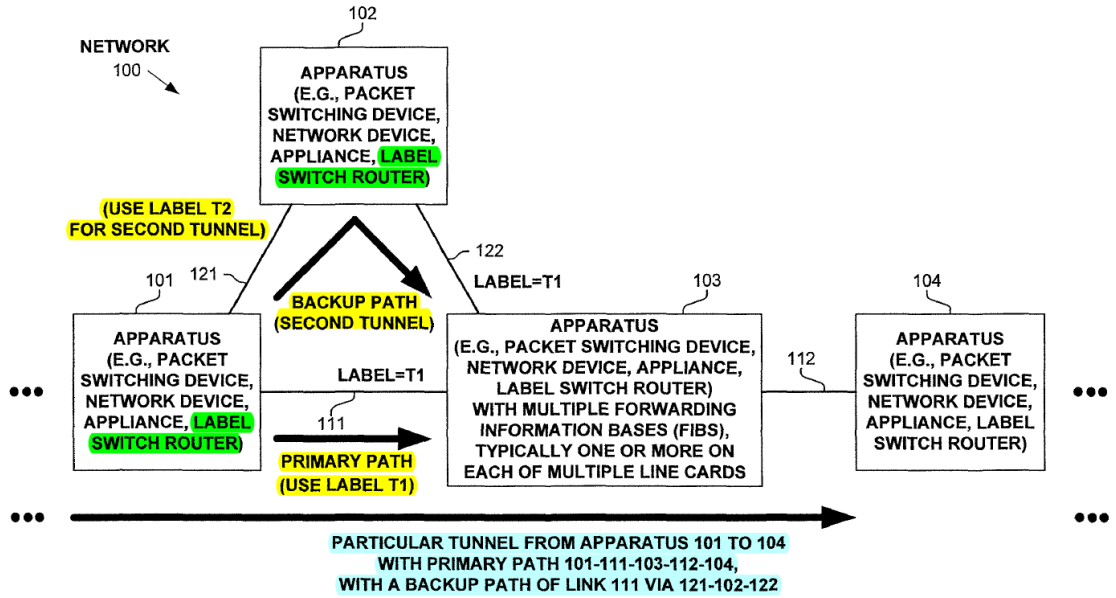
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | "FIG. 4 is a simplified block diagram of an example system 76 for providing another use case for pseudowire group labels. In this particular example, interface C fails (as shown at TPE2 64). Note that the same logical flow occurs in FIG. 4 in terms of the group signaling, as previously discussed. The group labels in two directions do not have to be the same, where the groupings for the messaging are not necessarily symmetrical. In this particular example, TPE2 64 sends a status for Group E with the corresponding group label (i.e., Group ID E for 300 pseudowires), where that message will have a Group Label E and a Group ID E. Hence, this particular signaling is indicative of 300 pseudowires failing in the network. SPE1 68 can send the status for Group F (where the Group ID F is associated with 300 pseudowires) to TPE1 62, where that message includes a Group Label F and a Group ID F." Rustogi, ¶ [0036].<br><br>"FIG. 5 is a simplified block diagram of an example system 80 for providing another use case for pseudowire group labels. In this particular example, interface D fails (as shown at TPE3 66), where all 700 pseudowires fail. In one implementation, TPE3 66 does not have a 700 pseudowire Group ID. Instead, the Group IDs can correspond to 200 and 500 pseudowires, when summed together account for the 700 pseudowires. In this particular example, TPE3 66 sends one message for Group I (representing 200 pseudowires) and another message for Group J (representing 500 pseudowires) to SPE1 68. In response, SPE1 68 sends a message for Group G (representing 200 pseudowires) and another message for Group H (representing 500 pseudowires). Again, the signaling being exchanged between these elements is minimal due to the effective grouping of pseudowires. SPE1 68 also sends a single message for Group I (associated with 200 pseudowires) and Group J (associated with 500 pseudowires) to TPE3 66, which is coupled to interface D. Group ID G is associated with 200 pseudowires, whereas Group ID H is representative of 500 pseudowires." Rustogi, ¶ [0037].<br><br>"FIG. 6 is a simplified table 74 illustrating an example set of pseudowire group provisioning parameters for TPE1 62, where these particular provisioning parameters could be relevant to the configuration of FIG. 3. At least in one generic sense, FIG. 2 can reflect one approach for mapping a PW group label to a PW Group ID, while FIGS. 3-5 can reflect a second |

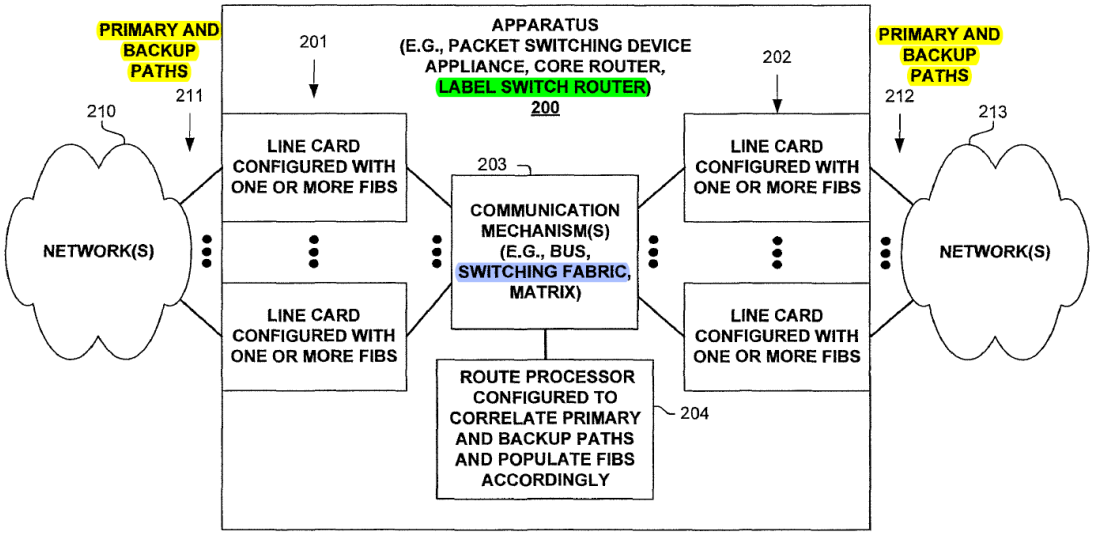
| No. | '821 Patent Claim 11 | The Reference |
|---|---|---|
| | | approach for such mappings, where table 74 is associated with that second approach." Rustogi, ¶ [0038].<br><br>"In particular, table 74 illustrates the mapping between SPE1 68 and TPE1 62. The first column represents the attachment circuit port (e.g., interface A, interface B, remote interface C on TPE2 64, and remote interface D on TPE3 66). Additionally, table 74 depicts a number of LSPs, a set of pseudowire grouping labels, and a set of pseudowire Group IDs. Note that the Group IDs are provided inside the pseudowire group labels in this example such that these two columns match in table 74. Additionally, note that table 74 is merely representing some of the possible characteristics in a single direction, where different constructs could be used in the reverse direction. Note that the provisioning as discussed herein can significantly reduce messaging such that these presented concepts offer increased scalability. This is due in part to the nominal processing that occurs in the network, in contrast to the processing required to evaluate a prolific amount of signaling messages associated with particular pseudowires. Additionally, the paradigm discussed herein can afford service providers an adequate amount of downtime after a failure has occurred in the network." Rustogi, ¶ [0039]. |

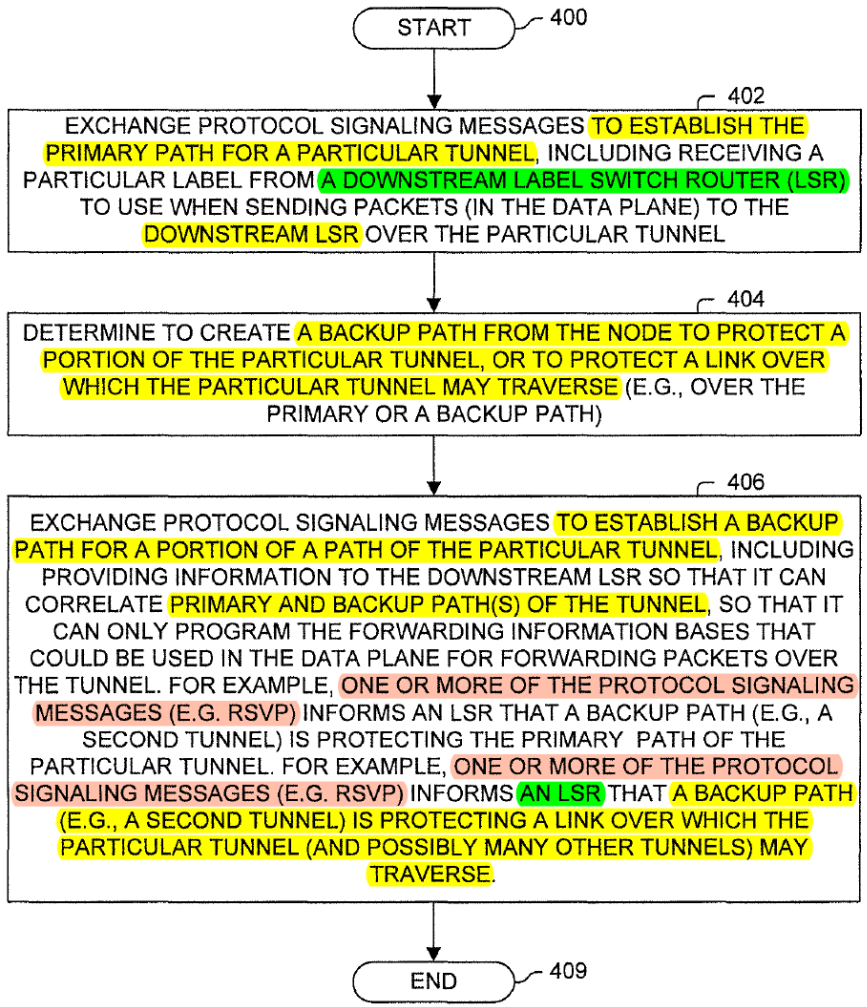
| No. | '821 Patent Claim 12 | The Reference |
|---|---|---|
| 12 | The method of claim 10, further comprising the step of configuring said working entity as revertive. | The Reference discloses the method of claim 10, further comprising the step of configuring said working entity as revertive.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

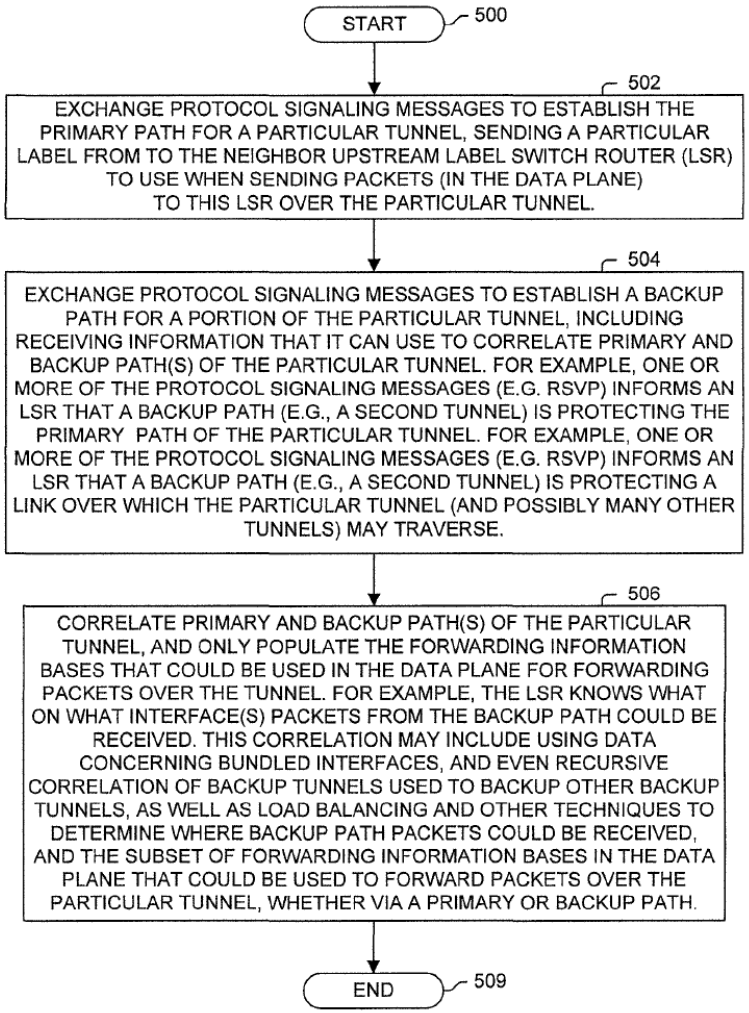
| No. | '821 Patent Claim 13 | The Reference |
|---|---|---|
| 13 | The method of claim 1, further comprising the step of: if said entity pair reselection results in both working and protection entities being replaced, sequentially replacing said working entity and said protection entity. | The Reference discloses the method of claim 1, further comprising the step of: if said entity pair reselection results in both working and protection entities being replaced, sequentially replacing said working entity and said protection entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| 14[preamble] | A system for selecting entities within an MPLS network, comprising: | The Reference discloses a system for selecting entities within an MPLS network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[preamble] and 1[a].<br><br>Cisco created and developed the MPLS and MPLS-TE standards and patented technology based on those standards *before* Orckit utilized such technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Filsfils<br>• Taylor<br>• Vasseur '879<br>• Rustogi |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | **Filsfils discloses**:<br>"In one embodiment, forwarding information bases (FIBs) are selectively populated in a packet switch. A packet switching device determines, based on one or more protocol signaling messages, a subset, which is less than all, on which FIBs a lookup operation may be performed for identifying forwarding information for a received particular packet. The packet switching device populates each of these FIBs, but not all of the FIBs of the packet switching device, with forwarding information corresponding to the particular forwarding value. Thus, FIB resources are consumed for only those FIBs which could actually be used, and not all of the FIBs, for forwarding packets in the data plane of the packet switching device, whether these packets are received on a primary or backup path." Filsfils, Abstract.<br>  <br><br>**FIGURE 1**<br>Filsfils, FIG. 1 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIGURE 2<br>Filsfils, FIG. 2 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 4**<br>Filsfils, FIG. 4 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 5**<br>Filsfils, FIG. 5. |

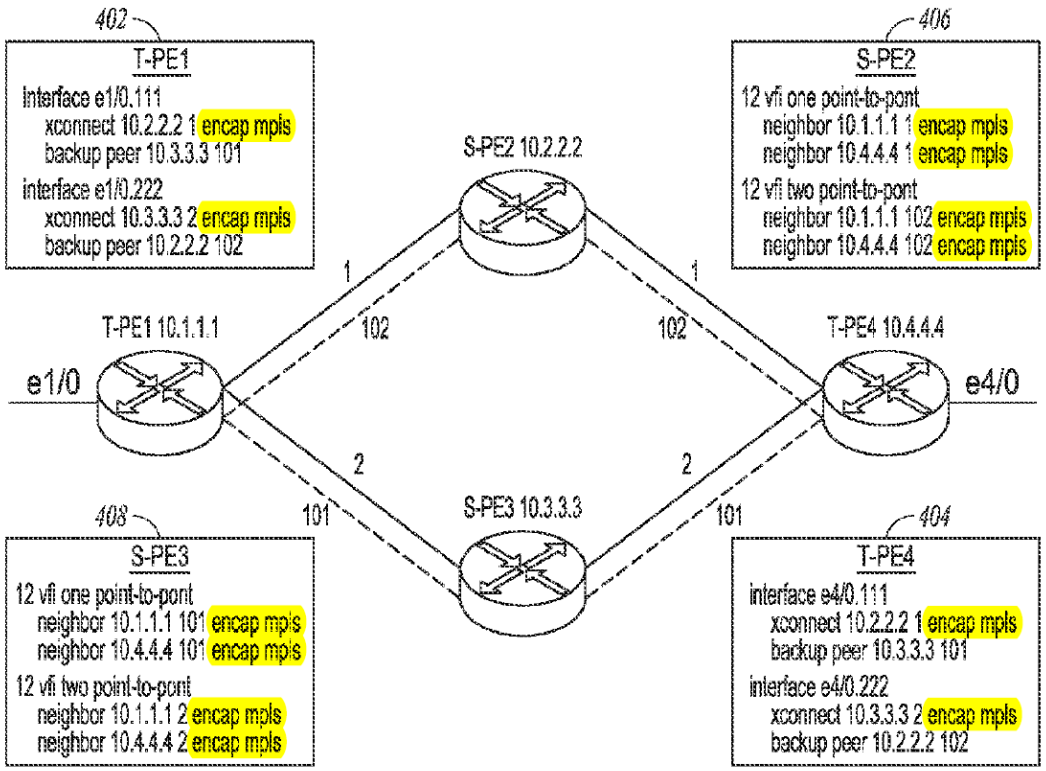
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology, including switching packets on labels especially in the core network using Multiprotocol Label Switching (MPLS)." Filsfils, 1:12-22.<br><br>"Tunnels, such as MPLS-TE (Traffic Engineering) and MPLS-TP (Transport Profile), are paths established through a network in order to transport packets efficiently through a label switched network. Fast Re-Route (FRR) is a technology that allows backup paths to be established in the network, which can be used in case of a problem with a primary path (original primary path or currently used backup path) of the tunnel. RFC 4090, entitled "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," provides an extension of the protocol signaling to establish backup label switched path (LSP) tunnels for local repair of LSP tunnels." Filsfils, 1:23-33.<br><br>"Expressly turning to the figures, FIG. 1 illustrates a network 100 operating according to one embodiment. Shown are four apparatus 101-104 (e.g., packet switching devices such as a label switch router, network device, and/or appliance). For explanation purposes, each of apparatus 101-104 will be referenced as a label switch router (LSR)." Filsfils, 5:41-46.<br><br>"As shown, a particular tunnel is established, using a signaling protocol and exchanging of protocol signaling messages. Note, LSR 101 may, or may not, be an endpoint of the particular tunnel (e.g., LSR 101 may be an intermediate LSR on the path of the particular tunnel). The primary path of the particular tunnel includes spans from LSR 101 via link 111 to LSR 103 and via link 112 to LSR 104. Note, LSR 104 may be an intermediate LSR on the path of the particular tunnel, or an endpoint of the particular tunnel. Further, for this example embodiment, LSR 103 signals LSR 101 to use label T1 at the top of the label stack in the header of a packet sent to it on the particular tunnel." Filsfils, 5:47-58.<br>"A second tunnel from LSR 101 via link 121 to LSR 102 and via link 122 to LSR 103 is similarly configured using a signaling protocol and exchanging of protocol signaling |

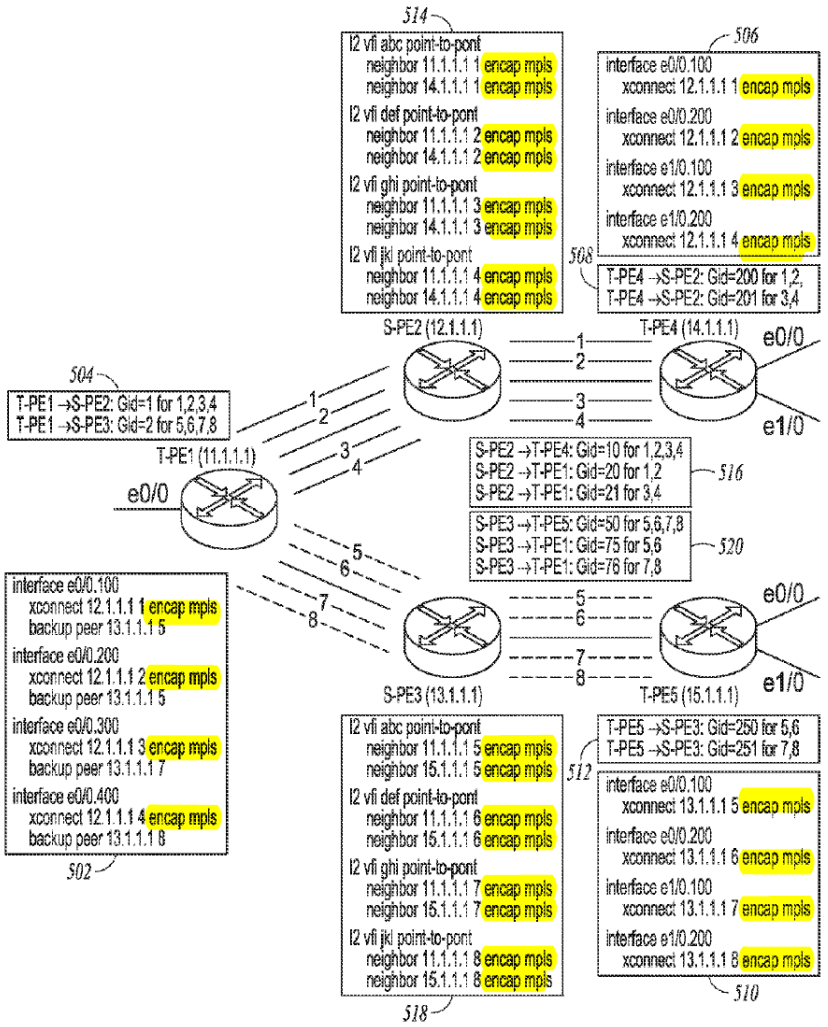
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | messages. For example purposes, LSR 102 signals LSR 101 to use label (T2) at the top of the label stack in the header of a packet sent to it on the second tunnel. In one embodiment, LSR 101 creates the second tunnel in response to determining, or being instructed to, create a backup path to protect link 111 and/or protect all or certain tunnels traversing link 111." Filsfils, 5:59-67.<br><br>"As shown in FIG. 1, link 111 (primary path of the particular tunnel and/or all or certain tunnels traversing link 111) is protected by LSR 101 using the second tunnel (backup path). When sending packets over the particular tunnel over link 111, LSR 101 includes label T1 at the top of the label stack of these packets. If link 111 cannot be used for communicating packets of the particular tunnel, LSR 101 sends packets over the backup path for the particular tunnel by sending packets to LSR 102, with these packets having a label stack including: label T2 followed by label T1. Thus, LSR 102 will receive these packets, pop the top label (T2) from the label stack of each of these packets, identify based on label T2 to send these packets to LSR 103. After popping the top label, the label at the top of the label stack of these packets is T1, which is the same label LSR 103 expects to receive for the particular tunnel. Therefore, these packets received with label T1 at the top of their label stack, are forwarded (after popping label T1 from their label stack) by LSR 103 over the particular tunnel to LSR 104." Filsfils, 6:6-24.<br><br>"One embodiment acquires such additional information by extending Resource Reservation Protocol (RSVP) to provide information which allows a packet switch to correlate primary and backup paths. Thus, a packet switch can use this additional information in determining which of its forwarding information bases (FIBs) could possibly be used in forwarding packets (e.g., in the data plane of the packet switch)." Filsfils, 6:51-57.<br><br>"In providing this additional information to LSR 103, one embodiment communicates an extended RSVP message (including a new or modified RSVP object) or other message to LSR 103 on the second tunnel. This messages designates one or more primary tunnels (e.g., label T1 in our example) and/or a link (e.g., link 111). As LSR 103 knows what interface that it received this message, LSR 103 knows that it must populate forwarding information for these primary tunnels, either specified (e.g., by a label such as T1), or all labels corresponding to |

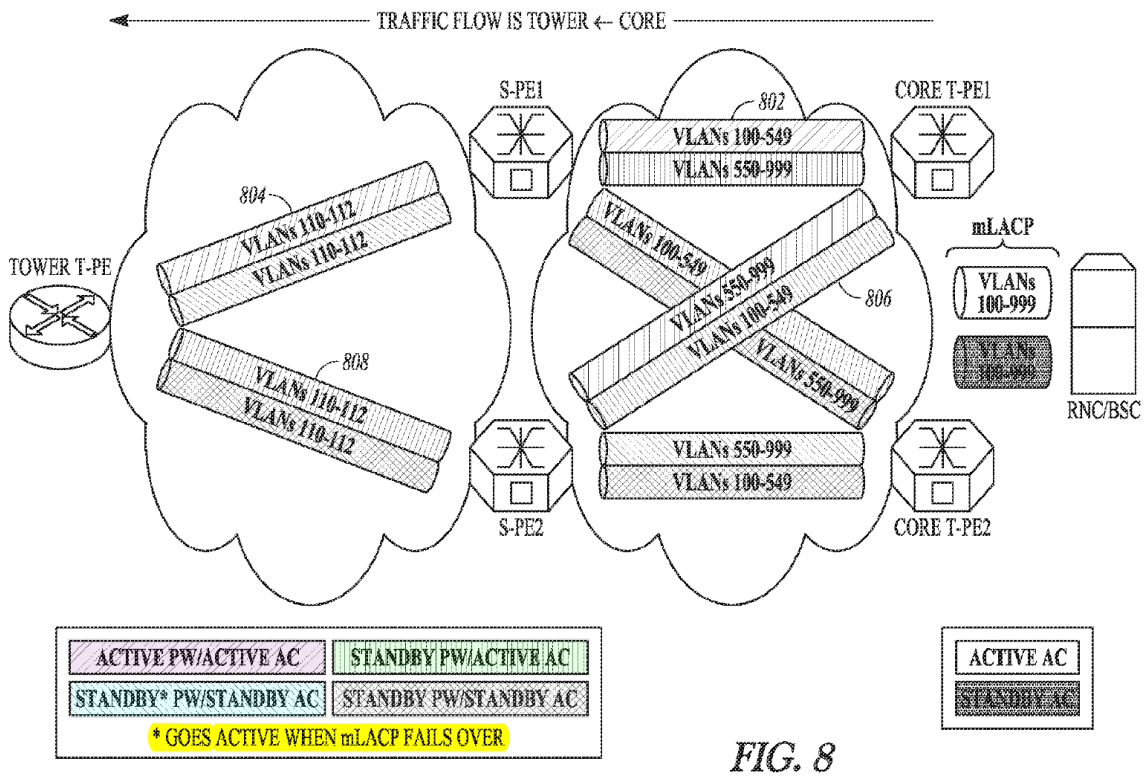
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | tunnels which could be received over link 111. In one embodiment, the extended RSVP or other message communicated to LSR 103 also includes an identification of the backup tunnel (e.g., T2) over which the RSVP or other message is being received, as the identification the tunnel over which a packet is received is often not communicated in a packet (e.g., in the case of Penultimate Hop Popping)." Filsfils, 7:19-34.<br><br>"As shown in FIG. 1, one embodiment includes apparatus 103, which populates less than all of its FIBs with forwarding information for a tunnel (although all FIBs may be populated for certain tunnels). One embodiment includes apparatus 101 and/or 102 which communicates, via a signaling protocol (e.g., an extension of RSVP, or using another protocol), information which allows apparatus 103 to determine the relationship between primary and backup paths, such that apparatus 104 can correlate this primary and backup path information (possibly also correlating backup path of backup path information, and/or bundled interfaces and/or bundled links) to identify a minimum subset of the FIBs that could possibly be used in forwarding packets of particular primary paths (e.g., tunnels)." Filsfils, 7:49-62<br><br>"Turning to FIG. 2, illustrates an apparatus 200 (e.g., packet switching devices such as a label switch router, network device, and/or appliance) operating in one embodiment. As shown, apparatus 200 includes line cards 201, 202 communicatively coupled via communication mechanism(s) 203 (e.g., bus, switching fabric, and/or matrix). Additionally, route processor 204 is configured to correlate primary and backup paths of tunnels, and to populate minimum subsets of FIBs with forwarding information for labels. Again, a minimum subset of FIBs for a particular path or label of the particular path is the set of FIBs that are determined to possibly be used in forwarding packets of a primary path, whether the label is received in a packet over the primary path or over a backup path, and possibly considering backup paths of a backup path and/or the possibly effect of bundled interfaces and/or bundled links." Filsfils, 7:63-8:11.<br><br>"As shown in FIG. 2, apparatus 200 is communicatively coupled via primary and backup paths 211, 212 to networks 210 and 213 (which could be the same network). As illustrated, each of line cards 201, 202 includes one or more FIBs. By correlating on which line card(s) 201, 202 and even within line cards 201, 202 that have multiple FIBs, primary and backup path(s) of tunnels, the number of FIB entries populated in apparatus 200 can typically be reduced, |

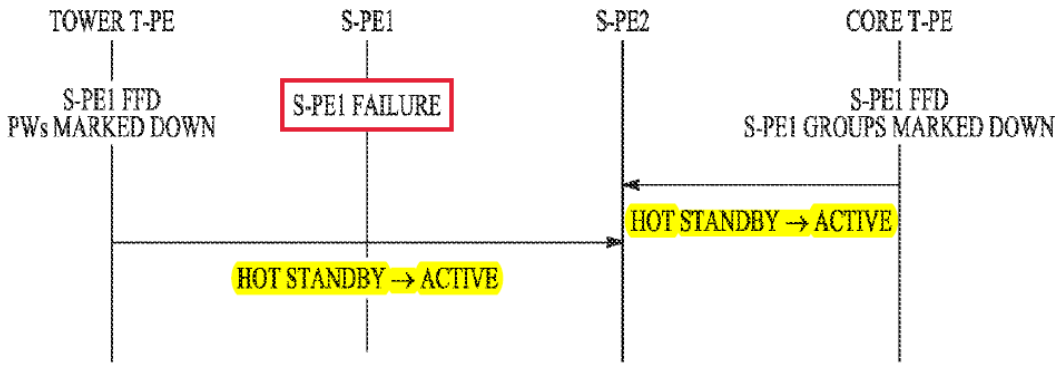
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | possibly significantly saving memory/storage resources and resources used to populate the FIBs." Filsfils, 8:12-21.<br><br>"FIG. 4 illustrates a process performed in one embodiment. Processing begins with process block 400. In process block 402, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes receiving a particular label for a downstream LSR to use when sending packets to the downstream LSR over the particular tunnel." Filsfils, 8:61-67.<br><br>"In process block 404, a determination is made to create a backup path from the node (e.g. the node performing these operations). This backup path may be used to protect one or more particular tunnels, and/or may be used to protect a link which is used to carry packet traffic of one or more tunnels." Filsfils, 9:1-5.<br><br>"In process block 406, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including providing information to the downstream LSR so that the downstream LSR can correlate primary and backup path(s) of the particular tunnel and substantially only program the FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:6-23.<br><br>"FIG. 5 illustrates a process performed in one embodiment. Processing begins with process block 500. In process block 502, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes sending a particular label for an upstream LSR to use when sending packets over the particular tunnel to this apparatus (e.g., an LSR performing these operations)." Filsfils, 9:26-32. |

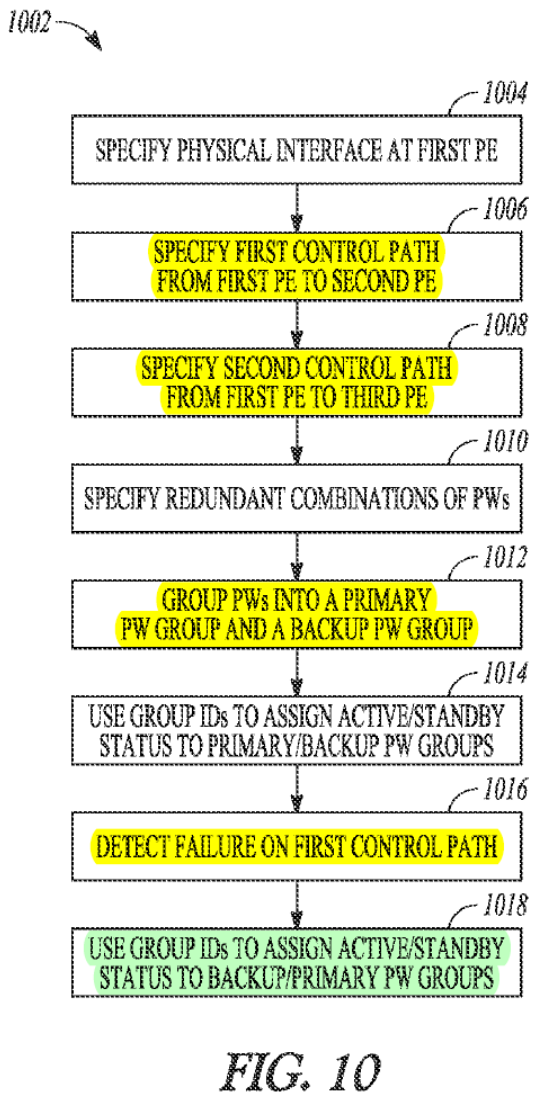
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "In process block 504, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including receiving information that the LSR can use to correlate primary and backup path(s) of the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:33-47.<br><br>"In process block 506, the primary and backup path(s) of the particular tunnel are correlated to identify the set of FIBs that could possibly be used in forwarding packets of the particular tunnel. Substantially only those FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel (either through a primary or backup path) are populated with the forwarding information (e.g., an entry corresponding to the label it advertised to use for the particular tunnel) for the particular tunnel. For example, the LSR knows what on what interface(s) packets from the backup path could be received. This correlation may include using data concerning bundled interfaces, and even recursive correlation of backup tunnels used to backup other backup tunnels, as well as load balancing and other techniques to determine where backup path packets could be received, and the subset of forwarding information bases in the data plane that could be used to forward packets over the tunnel, whether via a primary or backup path." Filsfils, 9:48-65.<br><br>**Taylor discloses**:<br>"Grouping pseudowires based on hardware interfaces and configured control paths enables improved pseudowire failover performance. Signaling status changes (e.g., from standby to active status) is facilitated by using group IDs for the pseudowire groups, thereby enabling improved failover performance when there is disruption in the network." Taylor, Abstract. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Taylor, FIG. 4 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 5*<br>Taylor, FIG. 5 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br>Taylor, FIG. 8 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Taylor, FIG. 9 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Taylor, FIG. 10 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The present disclosure relates generally to communication networks and more particularly to pseudowire configurations in communication networks." Taylor, 1:8-10.<br><br>"Virtual Private Network (VPN) services provide secure network connections between different locations. A company, for example, can use a VPN to provide secure connections between geographically dispersed sites that need to access the corporate network so that each customer edge (CE) end point or node can communicate directly and independently with all other CE nodes. Different types of VPNs have been classified by the network layer used to establish the connection between the customer and provider network. For example, Virtual Private LAN Service (VPLS) is an architecture that delivers a multipoint Layer 2 VPN (L2VPN) service that in all respects emulates an Ethernet Local Area Network (LAN) across a wide metropolitan geographic area. All services in a VPLS appear to be on the same LAN, regardless of location. In other words, with VPLS, customers can communicate as if they were connected via a private Ethernet segment, i.e., multipoint Ethernet LAN services." Taylor, 1:12-28.<br><br>"In this context, each CE device at a customer site is connected to the service provider network at a provider edge PE) device by an Attachment Circuit (AC) that provides the customer connection to a service provider network, that is, the connection between a CE node and its associated PE node. Within the provider network, each PE device includes a Virtual Switch Instance (VSI) that emulates an Ethernet bridge (i.e., switch) function in terms of Media Access Control (MAC) address learning and forwarding in order to facilitate the provisioning of a multipoint L2VPN. A pseudowire (PW) is a virtual connection between two PE devices that connect two attachment circuits. In the context of the VPLS service, a pseudowire can be thought of as a point-to-point virtual link for each offered service between a pair of VSIs. Therefore, if each VSI can be thought of as a virtual Ethernet switch for a given customer service instance, then each pseudowire can be thought of as a virtual link connecting these virtual switches to each other over a Packet Switched Network (PSN) for that service instance." Taylor, 1:29-47. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Since the failure of pseudowires obviously degrades network performance, some effort has been directed towards adding system redundancies including redundant pseudowires. However, the presence of redundant pseudowires alone is insufficient to improve overall failover performance, that is, the ability to switch over automatically to a redundant or backup system. Thus, there is a need for improved methods for managing pseudowires to facilitate pseudowire switching and enable improved failover performance." Taylor, 1:48-56.<br><br>"FIG. 4 shows details for PW connectivity in an exemplary network for an example embodiment." Taylor, 1:66-67.<br><br>"FIG. 5 shows details for PW grouping for an example embodiment." Taylor, 2:1-2.<br><br>"FIG. 8 shows an example network including redundant PW connections for an example embodiment." Taylor, 2:9-10.<br><br>"FIG. 9 shows an example sequence diagram for a failure mode related to the embodiment shown in FIG. 8." Taylor, 2:11-12.<br><br>"FIG. 10 shows a flowchart that illustrates a method of providing improved PW grouping according to an example embodiment." Taylor, 2:13-15.<br><br>"According to one embodiment, a method of providing improved pseudowire performance includes specifying a physical interface at a first PE node in a network, a first control path from the first PE node to a second PE node in the network, and a second control path from the first PE node to a third PE node in the network. With these specifications, the method then includes specifying redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node, and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Then these pseudowires can be grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. Group identifiers for the primary PW group and the backup PW group can then be used to assign an |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths. The method may then include detecting a failure on the first control path, and in response to the detected failure, using the group identifiers to assign the active status to the backup pseudowires and the standby status to the primary pseudowires." Taylor, 2:27-52.<br><br>"Pseudowires are used in pseudowire emulation edge-to-edge to provide a Layer 2 Virtual Private Network (L2VPN) connection. When large numbers (e.g., 4,000-6,000) of pseudowires are aggregated together on a single router, failure performance tends to be linear or O(n) where n is the number of pseudowires. While O(n) performance may be acceptable for small numbers of pseudowires, the effect on network outages can be increasingly undesirable as the number of pseudowires increases." Taylor, 2:54-62.<br><br>"For example, a cell-site router will typically start an approximately 2-minute procedure if contact with its controller, which is reached via a pseudowire, is lost for more than some threshold amount (e.g., between approximately 0.75 and 1.75 seconds in some cases). This can be a major impediment to the scalability of pseudowire deployments. These issues have become increasingly relevant as providers of Multiservice Broadband Networks (MBNs) are rapidly replacing or augmenting their traditional Synchronous Optical Networking (SONET) equipment with cheaper Ethernet equipment in the evolution towards a 4G (i.e., $4^{th}$ generation) network." Taylor, 2:63-3:7.<br><br>"One aspect of a solution to the problem of pseudowire failure is the deployment of redundant pseudowires. For example, redundant pseudowires have been used in the context of Multiprotocol Label Switching (MPLS) networks, which use a Label Distribution Protocol (LDP) to manage labels for forwarding traffic between routers. In this context, general requirements for redundancy schemes have been developed so that duplicate pseudowires are available when a given pseudowire fails (e.g., by using active/standby status indicators). In addition, more specific implementations for redundant pseudowires have also been developed." Taylor, 3:8-18. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 1 shows a reference network model 102 with applications to example embodiments disclosed herein. The reference network model 102 includes an aggregation network 104 of PE nodes and a distribution network 106 of PE nodes between a radio network controller (RNC) (or base station controller (BSC)) 108 on the core side of the model 102 and a radio tower 111 on the tower side of the model 102. Switching provider edge nodes S-PE1 and S-PE2 connect the two networks 104, 106. On the core side, two core terminating provider edges T-PE1 and T-PE2 connect to the RNC/BSC 108 through attachment circuits 110, 112. On the tower side, one tower terminating provider edge T-PE connects to the radio tower 111 through an attachment circuit 114." Taylor, 3:19-31.<br><br>"Additionally as noted in FIG. 1, peer-PE monitoring is carried out within each network 104, 106. That is, there is peer-PE monitoring between provider edges that share a segment, for example, by multi-hop bidirectional forwarding detection (BFD). Alternatively, peer monitoring can be accomplished by other means (e.g., MPLS-TP (Transport Protocol) keep-olives). This peer-PE monitoring is used to provide the mechanism for fast failure detection. Once a failure is detected, the network can react by "rerouting" the failed pseudowires to pre-provisioned backup paths and thus provide a minimal disruption in service to the end-user. This rerouting can be accomplished by LDP signaling between provider edges." Taylor, 3:32-44.<br><br>"The reference network model 102 may be considered as part of a larger hub-and-spoke model as shown in FIG. 2. A hub-and-spoke distribution model 202 includes a core network 204, distribution networks 206, and aggregation networks 208. Network elements including distribution nodes, aggregation nodes, and towers are also shown with nominal count values (e.g., 30 distribution nodes between the core network 204 and a distribution network 206). In this model 202, tower T-PEs are the spokes white core-PEs constitute the hub. Dozens to hundreds of tower T-PEs connect to a few S-PEs; these S-PEs are quite similar to ASBRs as they act as forwarders between the two distinct MPLS domains, providing isolation and, in the case of mobility, aggregation services. Typically, several aggregation networks 208 are connected to a single distribution network 206, eventually connecting the tower with the core router that connects the tower's ACs to the RNC/BSC. There are typically several distribution |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | networks in a Radio Access Network (RAN) connected to the service provider's core Internet Protocol (IP) network." Taylor, 4:11-30.<br><br>"With reference to FIG. 1, FIG. 3 shows a variety of failure modes encountered in the reference network model 102. Failure 302 of communications between tower T-PE and the S-PE can be detected via peer monitoring when both the S-PE and the T-PE are still active/alive. For example, this failure may be due to a toss of connectivity when the BFD session goes down. Failure 304 of S-PE1 can be due to a hardware failure, power outage, or the lack of BED-session maintenance capability (e.g., if the BFD-session hello timers cannot be serviced for the prescribed period of time). Failure 306 of communications between S-PE1 and core T-PE1 can be detected via peer monitoring when both S-PE1 and core T-PE1 are still active/alive. Failure 308 at core T-PE1 can be due to a hardware failure, power outage, or the lack of BFD-session maintenance capability." Taylor, 4:46-60.<br><br>"Pseudowire connectivity is further illustrated in FIG. 4 where the illustrated network includes four nodes: T-PE1 (10.1.1.1), S-PE2 (10.2.2.2), S-PE3 (10.3.3.3), and T-PEA (10.4.4.4). For the terminating nodes T-PE1 and T-PE4, specifications for VLANs (virtual Local Area Networks) connections (i.e., pseudowires) are shown using the Internet Operating System Command Line Interface (IOS CLI). The specification 402 for T-PE1 defines two VLANS as primary/backup combinations of virtual circuits for the network. The first three lines of the specification 402 define "VLAN 111" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 111" in the first line. The second line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE2 (10.12.2) with a virtual circuit Identification (VCID) set as VCID=1, and the third line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=101. The next three lines of the specification 402 define "VLAN 222" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 222" in the fourth line. The fifth line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=2, and the sixth line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE2 (10.2.2.2) with VCID=102. These virtual circuits, VCID=1, VCID=2, VCID=101 and VCID=102 are shown in the figure between T-PE1 and the S-PEs with a solid line for the primary circuits VCID=1 |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | and VCID=2 and a dashed line for the backup circuits VCID=101 and VCID=102." Taylor, 4:61-5:23.<br><br>"With respect to T-PE1 in FIG. 4, although "VLAN 111" and "VLAN 222" share the same hardware port, they do not share the same "control path disposition." That is, "VLAN 111" is primary to S-PE2 (VCID=1) and standby to S-PE3 (VCID=101), white "VLAN 222" has an opposite configuration since it is primary to S-PE3 (VCID=2) and standby to S-PE2 (VCID=102). As discussed below, certain embodiments group pseudowires according to "control path disposition" (e.g., xconnect configuration as well as the hardware interface in order to improve failover performance. That is, to deal with both hardware port failures and switching path failures, the grouping criteria also considers the cross connects. In this case, on T-PE1 as well as T-PE4, there would exist two groups: one for active to S-PE2 and standby to S-PE3 and another for active to S-PE3 and standby to S-PE2 (i.e., the inverse configuration)." Taylor, 5:61-6:9.<br><br>"First, local connectivity is characterized by local group identifications (Group-IDs), which depend on whether the allocation is done at a T-PE or S-PE. FIG. 5 shows an embodiment that illustrates an allocation of local group IDs in a network including terminating nodes T-PE1 (11.1.1.1), T-PE4 (14.1.1.1), and T-PE5 (15.1.1.1) and switching nodes S-PE2 (12.1.1.1) and S-PE3 (13.1.1.1). The specification 502 for T-PE1 determines corresponding local group IDs 504 based on the hardware interface and the control path. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, and the Group-ID=2 corresponds to VCID=5, VCID=6, VCID=7, and VCID=8." Taylor, 6:10-21.<br><br>"Local group IDs are maintained in a database so that pseudowire redundancy is also maintained. First, in a case without pseudowire redundancy, all the xconnect configurations from the same physical interface to the same peer are assigned the same local group ID. So, for example, in Ethernet cases all xconnect configurations under sub-interfaces of the same physical interface to the same peer will be assigned the same local group ID (e.g., e0/0 and e0/1 are sub-interfaces of e0). FIG. 6 shows a database representation for T-PE4 from FIG. 5. From the root node 602 for T-PE4, there is a first interface node 604 for e0/1 and a second interface node 606 for e1/0. The first interface node 604 is configured towards a single peer |

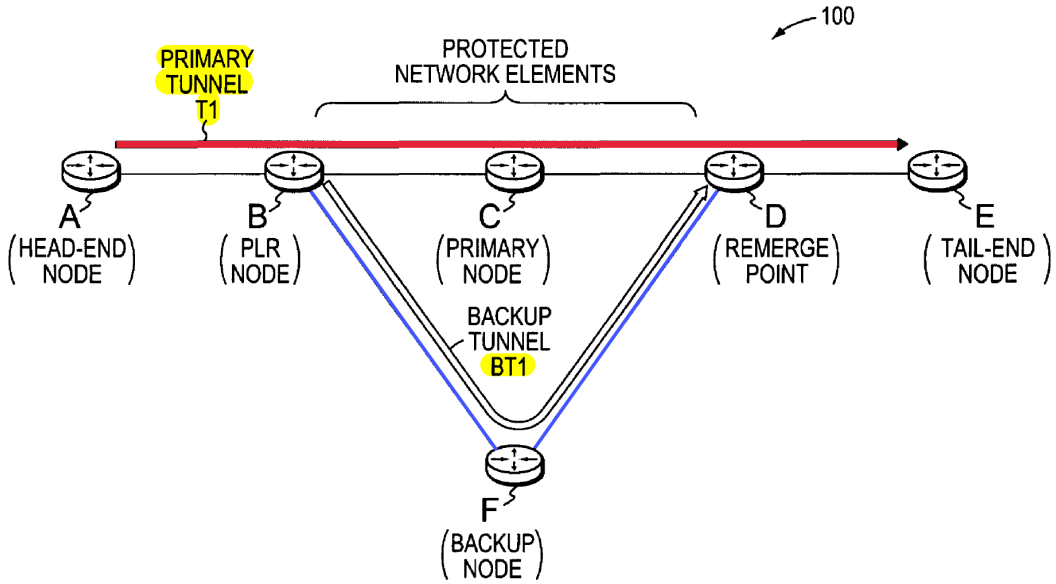
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | node (12.1.1.1) 608 and is thus assigned a single local group ID (Group-ID=200) 610. Similarly, the second interface node 606 is configured towards a single peer node (12.1.1.1) 612 and is thus assigned a single local group ID (Group-ID=201) 614. In this case from the assignment of local group IDs 508 in FIG. 5, Group-ID=200 corresponds to VCID=1 and VCID=2, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). Both pseudowires (VCID=1 and VCID=2) are assigned the same local group ID (Group-ID=200) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly, from the assignment of local group IDs 508 in FIG. 5, T-PE4 has VCID=3 and VCID=4 under the physical interface e1/0 going to the same peer S-PE2 (12.1.1.1), and the local Group ID (Group-ID=201) is assigned to these VCs." Taylor, 6:48-7:8.<br><br>"For the pseudowire redundancy case, a separate redundancy-group database is maintained by the xconnect application. This redundancy-group database contains the peer IDs in the group and the local group IDs advertised to them. This is needed to maintain a 1:1 mapping between the primary pseudowires and their corresponding backup pseudowires. FIG. 7 shows a database representation for T-PE1 from FIG. 5. From the root node 702 for T-PE1, there is an interface node 704 for e0/0 and a redundancy group node 706 that shows connections for configurations to a first peer node (12.1.1.1) 708, which is assigned a local group ID (Group-ID=1) 710, and a second peer node (13.1.1.1) 712, which is assigned a local group ID (Group-ID=2) 714. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). These pseudowires are assigned the same group ID (Group-ID=1) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly from the assignment of local group IDs 504 in FIG. 5, T-PE1 has VCID=5, VCID=6, VCID=7, and VCID=8 under the physical interface e0/0 going to another peer S-PE3 (13.1.1.1), and the local group ID (Group-ID=2) is assigned to these VCs. In this case these local group IDs are organized as a redundancy group 706." Taylor, 7:9-35. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Pseudowire grouping allows multiple pseudowires to be aggregated together when signaling either label withdrawals or status changes between segment end-point provider edges. This signaling can be carried out through LDP grouping TLV (Type Length Value). For example, when a PE node fails (e.g., failure 304 of S-PE1), aggregating the inter-segment PE signaling using the grouping TLV can provide significant scaling advantages. This allows all pseudowires sharing a physical port and PW configuration (e.g., xconnect configuration) to be signaled en masse between segment-adjacent provider edges." Taylor, 7:36-46.<br><br>"FIG. 8 shows an example based on FIG. 1 where VLAN ACs are shown as grouped by both port/HW-interface and pseudowire-class. The grouping criterion allows all "similar" pseudowires to be signaled together: All the grouped pseudowires share the same port and next-hop provider edge. Additionally, the figure contains many pseudowires, each grouped into a shaded tube. For example, the tube labeled "VLANs 100-549" contains 450 pseudowires grouped together. This figure depicts an incoming Ethernet comprised of 900 VLANs being segmented in two with 450 VLANs (100-549) active to S-PE1 while the other half of the VLANs (550-999) being active to S-PE2. This might be considered a type of manual load balancing. Furthermore, the aggregation network is only showing a single tower and the VLANs associated with it; other VLAN destinations are not shown in the figure." Taylor, 7:47-62.<br><br>"'VLANs 110-112' are active along a first pseudowire path 802 from Core T-PE1 to S-PE1 and a second pseudowire path 804 from S-PE1 to Tower T-PE. When a failure occurs at S-PE1 (e.g., as the switching node failure 304 shown in FIG. 3), then the standby pseudowires become active for 'VLANs 110-112' along a first pseudowire path 806 from Core T-PE1 to S-PE2 and a second pseudowire path 808 from S-PE2 to Tower T-PE." Taylor, 7:63-8:3.<br><br>"The standby pseudowires in FIG. 8 can be configured as HSPWs, a configuration that enables ACs to quickly failover to pre-provisioned pseudowires that are in active state but set to not-forwarding. Then when a failure occurs, switching over to these pre-provisioned HSPWs occurs quickly by switching from not-forwarding status to forwarding status." Taylor, 8:4-9. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 9 shows a UML (Unified Modeling Language) sequence diagram of signaling events related to the failover procedure illustrated in FIG. 8 for a switching node failure 304. The Tower T-PE and the Core T-PE independently detect a failure at S-PE1 (e.g., BFD forwarding with LDP signaling), and then mark down the status of the currently active pseudowires routed through S-PE1 and mark up the status of the standby pseudowires routed through S-PW2. Other failure modes shown in FIG. 3 can be handled similarly." Taylor, 8:10-18.<br><br>"In general, it is desirable for MPLS-based. Ethernet networks to react quickly to failures, so proactive detection mechanisms are employed in order to pick up system failures quickly. All proactive monitoring is typically done between PE peers on a single MPLS network. These provider edges on the edges of the MPLS network act similarly to an Autonomous System Boundary Router (ASBR). As a result, related embodiments detect control path failures, which may not be the same as pseudowire data path failures. That is, the data packets and control packets may take different paths between provider edges in a MPLS network although typically these paths are coincident. Thus, when the control and data paths are not coincident, if the control path fails, then all pseudowires utilizing the control path are marked as failed. As a corollary, if the data path fails and the control path remains healthy, then failure will not be detected from monitoring the control path." Taylor, 8:19-35.<br><br>"A failure of a monitored provider edge initiates a switchover of all active pseudowires using the failing provider edge to their configured HSPWs (if they exist). Grouping can greatly reduce the number of messages needed between provider edges (Inter-PE Aggregation) and within a single provider edge (Intra-PE Aggregation). Furthermore, the MPLS network itself may be internally resilient deploying technologies such as, but not limited to, MPLS-TE (MPLS Traffic Engineering) and ERR (Fast Reroute). The paths across the MPLS network may recover quickly and might not trip the fault-monitoring systems." Taylor, 8:36-46.<br><br>"With reference to the above discussion, FIG. 10 shows a method 1002 of providing improved PW grouping according to an example embodiment. In a first operation 1004 of the method 1002, a physical interface is specified at a first PE node in a network. In a second operation 1006, a first control path is specified from the first PE node to a second PE node in the network. In a third operation 1008, a second control path is specified from the first PE node to a third |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | PE node in the network. These control paths related to a common physical interface can be used to characterize redundant pairs of pseudowires." Taylor, 8:48-58.<br><br>"In a fourth operation 1010, redundant combinations of pseudowires are specified, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Although a redundant combination may relate a single backup pseudowire to a given primary pseudowire, in some cases multiple backup pseudowires will be related to a given primary pseudowire for increased redundancy. In a fifth operation 1012, these pseudowires are grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. In a sixth operation 1014, group identifiers for the primary PW group and the backup PW group are used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 8:59-9:11.<br><br>"In an optional seventh operation 1016, a failure may be detected on the first control path, and in an optional eighth operation 1018, in response to the detected failure, the group identifiers may be used to assign the active status to the backup pseudowires and the standby status to the primary pseudowires. For example, the failure on the first control path may be detected by using BED packet streams between PE nodes of the network. Then the detected failure can be signaled to PE nodes in the network by sending LDP status updates between PE nodes in the network. Then, after receiving the failure detection signals, the group identifiers can be used again to assign the active status to the backup pseudowires and the standby status to the primary pseudowires by sending LDP status updates between PE nodes in the network." Taylor, 9:12-26.<br><br>"Typically the network in is an MPLS network and the PE nodes are routers that provide network services to connected CE nodes of a customer network. In general, each control path is an Internet Protocol (IP) routing path between PE nodes in the network and each data path is a label switched path (LSP) between PE nodes in the network." Taylor, 9:27-32. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 11 shows a schematic representation of an apparatus 1102, in accordance with an example embodiment. For example, the apparatus 1102 may be used to implement the method 1002 of providing improved pseudowire grouping as described above with reference to FIG. 10. The apparatus 1102 is shown to include a processing system 1104 that may be implemented on a server, client, or other processing device that includes an operating system 1106 for executing software instructions." Taylor, 10:2-10.<br><br>"In accordance with an example embodiment, the apparatus 1102 includes a PW management module 1108 that includes a first specification module 1110, a second specification module 1112, third specification module 1114, a fourth specification module 1116, a grouping module 1118, and an assignment module 1120. The first specification module 1110 operates to specify a physical interface at a first PE node in a network. The second specification module 1112 operates to specify a first control path from the first PE node to a second PE node in the network. The third specification module 1114 operates to specify a second control path from the first PE node to a third PE node in the network. The fourth specification module 1116 operates to specify redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node." Taylor, 10:11-29.<br><br>"The grouping module 1118 operates to group the pseudowires into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. The assignment module 1120 operates to use group identifiers for the PW groups to assign an active status to the primary pseudowires and a standby status to the backup pseudowires. The active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 10:30-38. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | **Vasseur '879 discloses**:<br>"A technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, Abstract.<br>  <br><br>FIG. 1<br>Vasseur '879, FIG. 1 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|

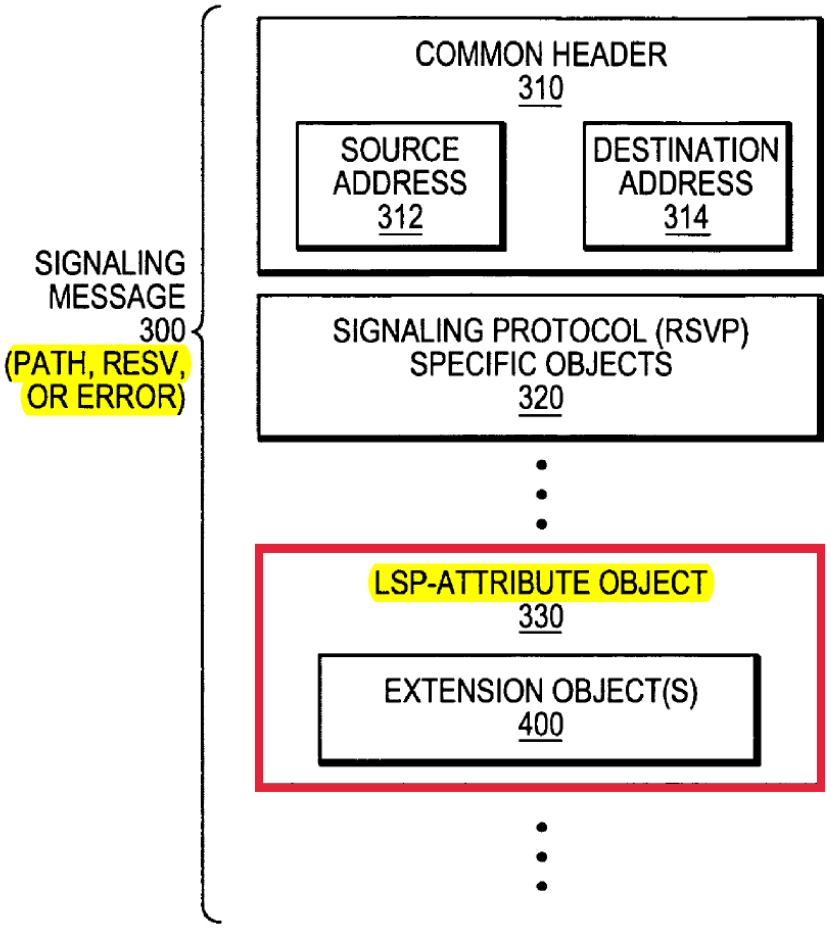
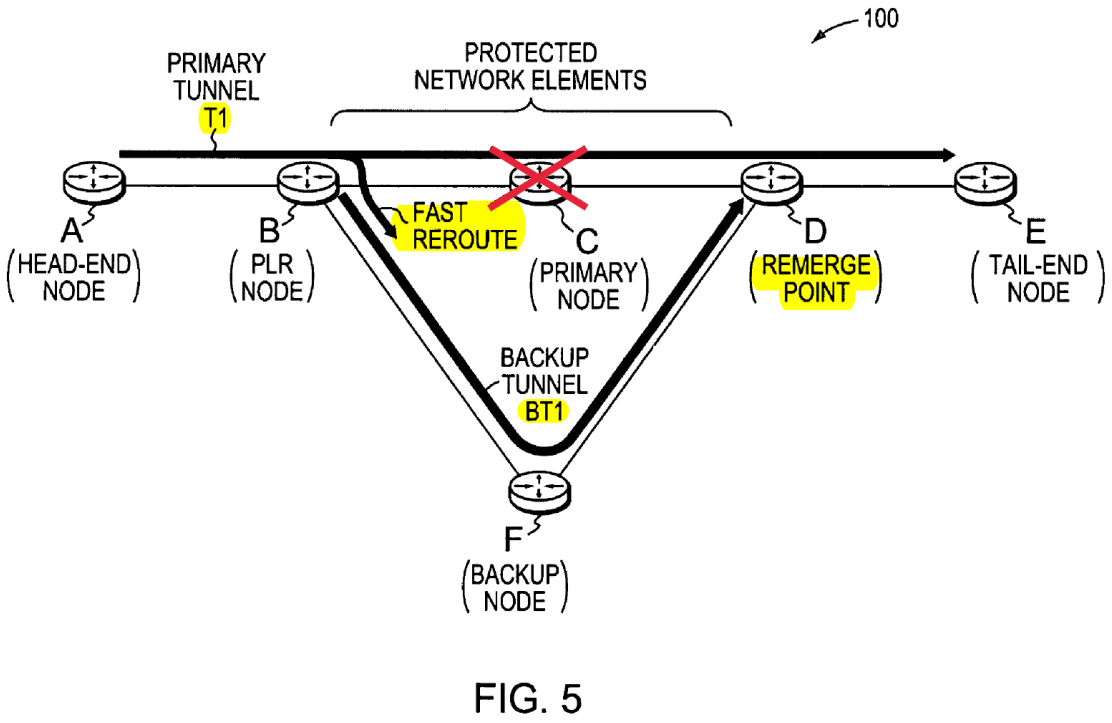
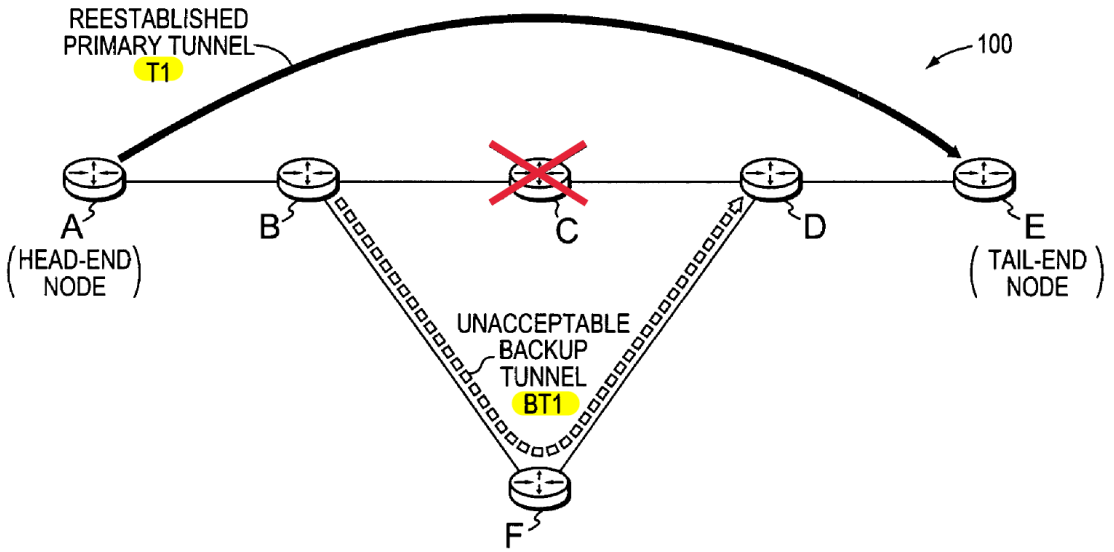
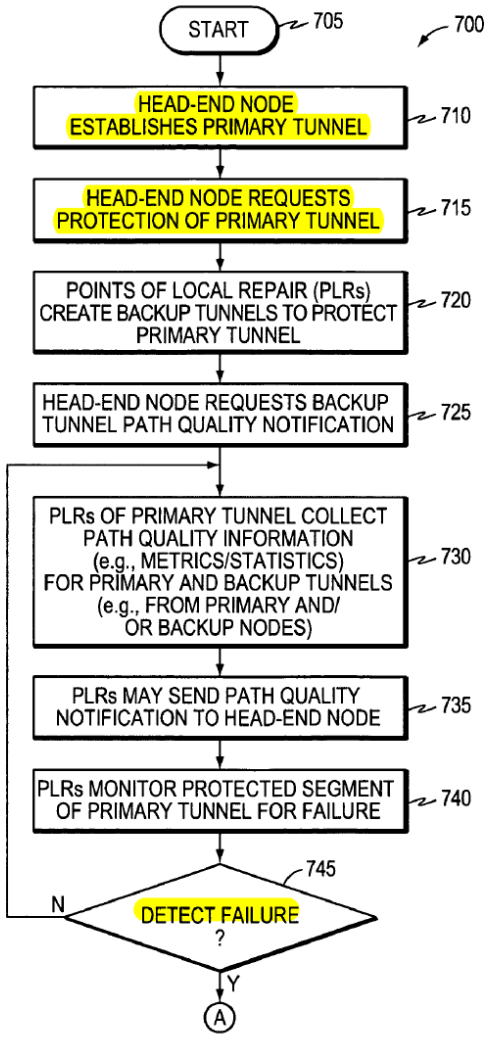


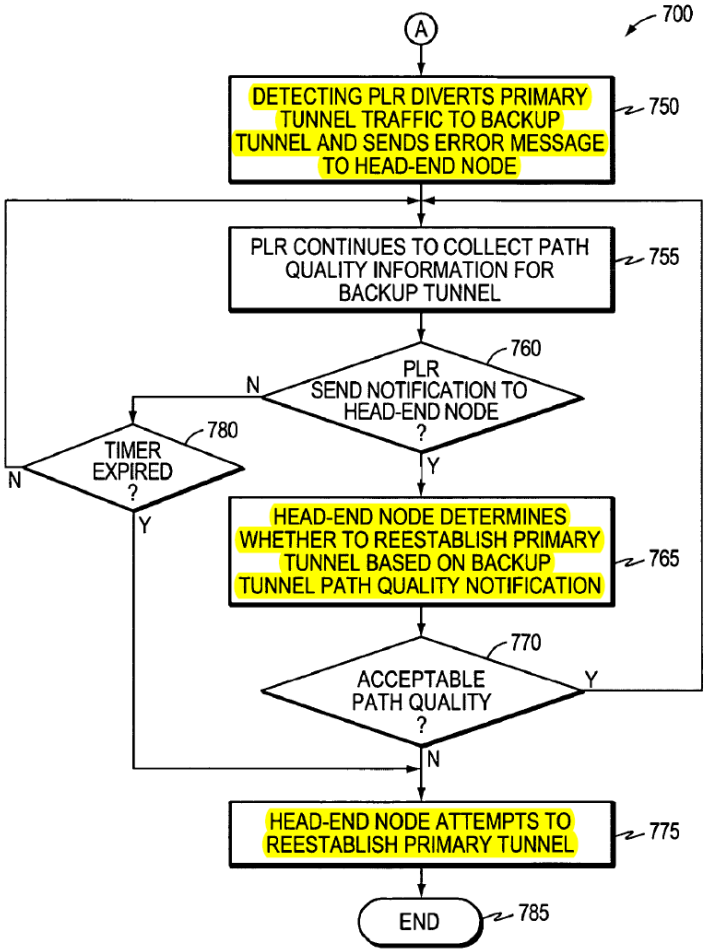

FIG. 3

Vasseur '879, FIG. 3 (annotated).

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Vasseur '879, FIG. 5 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br>Vasseur '879, FIG. 6 (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 7A<br>Vasseur '879, FIG. 7A (annotated). |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 7B<br>Vasseur '879, FIG. 7B. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "The present invention relates to computer networks and more particularly to dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 1:10-14.<br><br>"Since management of interconnected computer networks can prove burdensome, smaller groups of computer networks may be maintained as routing domains or autonomous systems. The networks within an autonomous system (AS) are typically coupled together by conventional "intradomain" routers configured to execute intradomain routing protocols, and are generally subject to a common authority. To improve routing scalability, a service provider (e.g., an ISP) may divide an AS into multiple "areas." It may be desirable, however, to increase the number of nodes capable of exchanging data; in this case, interdomain routers executing interdomain routing protocols are used to interconnect nodes of the various ASes. Moreover, it may be desirable to interconnect various ASes that operate under different administrative domains. As used herein, an AS or an area is generally referred to as a "domain," and a router that interconnects different domains together is generally referred to as a 'border router.'" Vasseur '879, 1:40-56.<br><br>"An example of an interdomain routing protocol is the Border Gateway Protocol version 4 (BGP), which performs routing between domains (ASes) by exchanging routing and reachability information among neighboring interdomain routers of the systems. An adjacency is a relationship formed between selected neighboring (peer) routers for the purpose of exchanging routing information messages and abstracting the network topology. The routing information exchanged by BGP peer routers typically includes destination address prefixes, i.e., the portions of destination addresses used by the routing protocol to render routing ("next hop") decisions. Examples of such destination addresses include IP version 4 (IPv4) and version 6 (IPv6) addresses. BGP generally operates over a reliable transport protocol, such as TCP, to establish a TCP connection/session. The BGP protocol is well known and generally described in Request for Comments (RFC) 1771, entitled A Border Gateway Protocol 4 (BGP-4), published March 1995." Vasseur '879, 1:57-2:7. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Examples of an intradomain routing protocol, or an interior gateway protocol (IGP), are the Open Shortest Path First (OSPF) routing protocol and the Intermediate-System-to-Intermediate-System (IS-IS) routing protocol. The OSPF and IS-IS protocols are based on link-state technology and, therefore, are commonly referred to as link-state routing protocols. Link-state protocols define the manner with which routing information and network-topology information are exchanged and processed in a domain. This information is generally directed to an intradomain router's local state (e.g., the router's usable interfaces and reachable neighbors or adjacencies). The OSPF protocol is described in RFC 2328, entitled OSPF Version 2, dated April 1998 and the IS-IS protocol used in the context of IP is described in RFC 1195, entitled Use of OSI IS-IS for routing in TCP/IP and Dual Environments, dated December 1990, both of which are hereby incorporated by reference." Vasseur '879, 2:8-24.<br><br>"An intermediate network node often stores its routing information in a routing table maintained and managed by a routing information base (RIB). The routing table is a searchable data structure in which network addresses are mapped to their associated routing information. However, those skilled in the art will understand that the routing table need not be organized as a table, and alternatively may be another type of searchable data structure. Although the intermediate network node's routing table may be configured with a predetermined set of routing information, the node also may dynamically acquire ("learn") network routing information as it sends and receives data packets. When a packet is received at the intermediate network node, the packet's destination address (e.g., stored in a header of the packet) may be used to identify a routing table entry containing routing information associated with the received packet. Among other things, the packet's routing information indicates the packet's next-hop address." Vasseur '879, 2:25-41.<br><br>"Multi-Protocol Label Switching (MPLS) Traffic Engineering has been developed to meet data networking requirements such as guaranteed available bandwidth or fast restoration. MPLS Traffic Engineering exploits modem label switching techniques to build guaranteed bandwidth end-to-end tunnels through an IP/MPLS network of label switched routers (LSRs). These tunnels are a type of label switched path (LSP) and thus are generally referred to as MPLS Traffic Engineering (TE) LSPs. Examples of MPLS TE can be found in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels dated December 2001, RFC 3784 |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | entitled Intermediate-System-to-Intermediate-System (IS-IS) Extensions for Traffic Engineering (TE) dated June 2004, and RFC 3630, entitled Traffic Engineering (TE) Extensions to OSPF Version 2 dated September 2003, the contents of all of which are hereby incorporated by reference in their entirety." Vasseur '879, 2:58-3:6.<br><br>"Establishment of an MPLS TE-LSP from a head-end LSR to a tail-end LSR involves computation of a path through a network of LSRs. Optimally, the computed path is the "shortest" path, as measured in some metric, that satisfies all relevant LSP Traffic Engineering constraints such as e.g., required bandwidth, "affinities" (administrative constraints to avoid or include certain links), etc. Path computation can either be performed by the head-end LSR or by some other entity operating as a path computation element (PCE) not co-located on the head-end LSR. The head-end LSR (or a PCE) exploits its knowledge of network topology and resources available on each link to perform the path computation according to the LSP Traffic Engineering constraints. Various path computation methodologies are available including CSPF (constrained shortest path first). MPLS TE-LSPs can be configured within a single domain, e.g., area, level, or AS, or may also span multiple domains, e.g., areas, levels, or ASes." Vasseur '879, 3:7-24.<br><br>"The PCE is an entity having the capability to compute paths between any nodes of which the PCE is aware in an AS or area. PCEs are especially useful in that they are more cognizant of network traffic and path selection within their AS or area, and thus may be used for more optimal path computation. A head-end LSR may further operate as a path computation client (PCC) configured to send a path computation request to the PCE, and receive a response with the computed path, potentially taking into consideration other path computation requests from other PCCs. It is important to note that when one PCE sends a request to another PCE, it acts as a PCC." Vasseur '879, 3:25-36.<br><br>"Some applications may incorporate unidirectional data flows configured to transfer time-sensitive traffic from a source (sender) in a computer network to a destination (receiver) in the network in accordance with a certain "quality of service" (QoS). Here, network resources may be reserved for the unidirectional flow to ensure that the QoS associated with the data flow is maintained. The Resource ReSerVation Protocol (RSVP) is a network-control protocol that |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | enables applications to reserve resources in order to obtain special QoS for their data flows. RSVP works in conjunction with routing protocols to, e.g., reserve resources for a data flow in a computer network in order to establish a level of QoS required by the data flow. RSVP is defined in R. Braden, et al., Resource ReSerVation Protocol (RSVP), RFC 2205, the contents of which are hereby incorporated by reference in its entirety. In the case of traffic engineering applications, RSVP signaling (with Traffic Engineering extensions) is used to establish a TE-LSP and to convey various TE-LSP attributes to routers, such as border routers, along the TE-LSP obeying the set of required constraints whose path may have been computed by various means." Vasseur '879, 3:37-57.<br><br>"Generally, a tunnel is a logical structure that encapsulates a packet (a header and data) of one protocol inside a data field of another protocol packet with a new header. In this manner, the encapsulated data may be transmitted through networks that it would otherwise not be capable of traversing. More importantly, a tunnel creates a transparent virtual network link between two network nodes that is generally unaffected by physical network links or devices (i.e., the physical network links or devices merely forward the encapsulated packet based on the new header). While one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 3:58-4:3.<br><br>"Occasionally, a network element (e.g., a node or link) will fail, causing redirection of the traffic that originally traversed the failed network element to other network elements that bypass the failure. Generally, notice of this failure is relayed to the nodes in the network through an advertisement of the new network topology, e.g., an IGP or BGP Advertisement, and routing tables are updated to avoid the failure accordingly. Reconfiguring a network in response to a network element failure using, e.g., pure IP rerouting, can be time consuming. Many recovery techniques, however, are available to provide fast recovery and/or network configuration in the event of a network element failure, including, inter alia, "Fast Reroute", e.g., MPLS TE Fast Reroute. An example of MPLS TE Fast Reroute is described in Pan, et al., Fast Reroute Extensions to RSVP-TE for LSP Tunnels, RFC 4090, May 2005, which is hereby incorporated by reference as though fully set forth herein." Vasseur '879, 4:4-21. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Fast Reroute (or FRR) has been widely deployed to protect against network element failures, where "backup tunnels" are created to bypass one or more protected network elements (e.g., links, shared risk link groups (SRLGs), and nodes). When the network element fails, traffic is quickly diverted ("Fast Rerouted") over a backup tunnel to bypass the failed element, or more particularly, in the case of MPLS, a set of primary TE-LSPs (tunnels) is quickly diverted. Specifically, the point of local repair (PLR) node configured to reroute the traffic inserts ("pushes") a new label for the backup tunnel, and the traffic is diverted accordingly. Once the failed element is bypassed, the backup tunnel label is removed ("popped"), and the traffic is routed along the original path according to the next label (e.g., that of the original TE-LSP). Notably, the backup tunnel, in addition to bypassing the failed element along a protected primary TE-LSP, also intersects the primary TE-LSP, i.e., it begins and ends at nodes along the protected primary TE-LSP." Vasseur '879, 4:22-39.<br><br>"To offer maximum protection, e.g., guaranteed bandwidth, during Fast Reroute, backup tunnels may reserve a configurable amount of bandwidth to ensure that Fast Rerouted traffic from the primary tunnel has a reserved path to follow. For example, the bandwidth reserved for the primary tunnel may also be reserved on the backup tunnel. While this approach provides maximum protection, it also requires a non-negligible amount of network resources (e.g., capacity/bandwidth) and may increase operational complexity." Vasseur '879, 4:40-48.<br><br>"Certain techniques are available to efficiently minimize the amount of resources required by the establishment and maintenance of the backup tunnels for Fast Reroute. One such technique is to create zero-bandwidth ("0-BW") backup tunnels (i.e., tunnels that reserve no bandwidth) to protect non-0-BW primary tunnels. This "best effort" approach does not guarantee that the path followed by the backup tunnel will have enough bandwidth to support the diverted primary tunnel at the time of failure without QoS degradation, however in many situations the path quality of the backup tunnel is sufficient. For instance, if the network is not overly congested, or the backup tunnel follows a non-congested path, there may be enough available bandwidth along the backup tunnel to support the newly rerouted traffic. Also, because primary tunnels often reserve bandwidth in response to "peak" traffic utilization, the amount of traffic over the primary tunnel at the time of failure may be far less than the reserved bandwidth (e.g., at "off-peak" times). Notably, those skilled in the art will understand that in |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | the absence of the above exceptions, a 0-BW backup tunnel may have unacceptable bandwidth (e.g., affecting path quality) to support the diverted traffic." Vasseur '879, 4:49-5:2.<br><br>"Currently, head-end nodes (LSRs) may be configured to systematically reroute the primary tunnels affected by the network element failure (e.g., diverted primary tunnels), especially in the case with 0-BW backup tunnels, such as, e.g., by reestablishing a new primary tunnel that follows a path excluding the failed network element. In particular, 0-BW backup tunnels represent a best effort attempt to allow the head-end node to more gracefully reestablish the primary tunnel in response to a failure, since the backup tunnels may not be able to support the diverted traffic without QoS degradation. The systematic reestablishing may potentially result in the reestablishment of a large number of primary tunnels (e.g., up to 3000 for a single network element failure in today's networks). Notably, reestablishing diverted primary tunnels may be undesirable for the network, such as by creating traffic churn, jitter, control plane overloads, etc., as will be understood by those skilled in the art. However, as noted above, there are situations where the backup tunnel may provide acceptable bandwidth, at least, for example, for a period of time (e.g., possibly short) until the failed network element is restored. In these situations, then, it may have been unnecessary to reestablish the diverted primary tunnels. There remains a need, therefore, for a technique that dynamically determines whether to reestablish a diverted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 5:3-28.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 5:32-47. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as Resource ReSerVation Protocol (RSVP) Traffic Engineering (TE) signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or type/length/value (TLV) encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object of the RSVP messages to convey the path quality notification." Vasseur '879, 5:48-58.<br><br>"In accordance with one aspect of the present invention, the head-end node requests the establishment of the primary tunnel (e.g., a TE-Label Switched Path, TE-LSP), along with a request for Fast Reroute protection of one or more network elements (e.g., with zero-bandwidth, 0-BW backup tunnels) at a PLR node. In addition, the head-end node may include a request for backup tunnel path quality notification, such as, e.g., through the use of the novel Feedback sub-object. The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and sends an error message (e.g., an RSVP PathErr) to the head end node indicating the "Fast Rerouting" of the primary tunnel." Vasseur '879, 5:59-6:6.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, the PLR node may collect metrics/statistics (e.g., packet drops, path cost, jitter, etc.) of the primary and/or backup tunnels. Once the primary tunnel is Fast Rerouted, the PLR node continues to collect metrics of the backup tunnel, and may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. Notably, the PLR node may be configured to send path quality notifications to the head-end node once, continually, periodically, in response to configurable changes in path quality, etc. Also, as in the case where multiple primary tunnels are Fast Rerouted, the path quality notification may include an indication of which Fast Rerouted primary tunnels in particular have been effected by the changed path quality." Vasseur '879, 6:7-23. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "In accordance with yet another aspect of the present invention, upon receiving the error message (PathErr), the head-end node may wait for the path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. Notably, in the event the head-end node does not receive a path quality notification for the backup tunnel (e.g., within a configurable time limit), the head-end node may attempt to reestablish the new primary tunnel. Moreover, where the head-end node has multiple primary tunnels being Fast Rerouted, a configurable subset of the primary tunnels may be reestablished, e.g., to reduce congestion of the backup tunnels, and/or to limit the number of reestablished primary tunnels within a given period of time." Vasseur '879, 6:24-43.<br><br>"Advantageously, the novel technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. By providing the head-end node of the primary tunnel with path quality feedback of the backup tunnel, the novel technique avoids reestablishing a potentially large number of tunnels over one or more alternate paths after a failure (and Fast Reroute) if the backup tunnels have acceptable path quality. In particular, the backup tunnels, e.g., 0-BW backup tunnels, may not be congested or subsequently burdened by the Fast Rerouted traffic of the primary tunnel. Also, the failed network element (thus the primary tunnel) may be quickly restored; therefore by not reestablishing the primary tunnel, network jitter, churn, etc., may be avoided. Further, the dynamic nature of the novel technique alleviates the need for cumbersome manual configuration." Vasseur '879, 6:44-59.<br><br>"FIG. 3 is schematic block diagram of an exemplary signaling (RSVP) message that may be advantageously used with the present invention." Vasseur '879, 7:6-8.<br><br>"FIG. 5 is a schematic block diagram of the computer network in FIG. 1 showing Fast Reroute protection of a primary tunnel using a backup tunnel in accordance with the present invention." Vasseur '879, 7:12-15. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 6 is a schematic block diagram of the computer network in FIG. 5 showing an unacceptable backup tunnel path quality and resultant reestablishing of the primary tunnel in accordance with the present invention." Vasseur '879, 7:16-19.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention." Vasseur '879, 7:20-24.<br><br>"FIG. 1 is a schematic block diagram of an exemplary computer network 100 comprising a plurality of nodes A-F, such as routers or other network devices, interconnected as shown. The nodes may be a part of one or more autonomous systems, routing domains, or other networks or subnetworks. For instance, routers A and E may be provider edge (PE) devices of a provider network, (e.g., a service provider network) that are interconnected to one or more customer networks through customer edge (CE) devices (not shown, while the remaining nodes B-D and F may be core provider (P) devices, as will be understood by those skilled in the art. Those skilled in the art will also understand that the nodes A-F may be any nodes within any arrangement of computer networks, and that the view shown herein is merely an example. For example, the nodes may be configured as connections to/from one or more virtual private networks (VPNs), as will be understood by those skilled in the art. These examples are merely representative. Those skilled in the art will understand that any number of routers, nodes, links, etc. may be used in the computer network 100 and connected in a variety of ways, and that the view shown herein is for simplicity." Vasseur '879, 7:29-49.<br><br>"Data packets may be exchanged among the computer network 100 using predefined network communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Asynchronous Transfer Mode (ATM) protocol, Frame Relay protocol, Internet Packet Exchange (IPX) protocol, etc. Routing information may be distributed among the routers of the computer network using predetermined Interior Gateway Protocols (IGPs), such as conventional distance-vector protocols or, illustratively, link-state protocols, through the use of IGP Advertisements." Vasseur '879, 7:50-60. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 2 is a schematic block diagram of an exemplary router 200 that may be advantageously used with the present invention, e.g., as an edge router or a core router. The router comprises a plurality of network interfaces 210, a processor 220, and a memory 240 interconnected by a system bus 250. The network interfaces 210 contain the mechanical, electrical and signaling circuitry for communicating data over physical links coupled to the network 100. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols, including, inter alia, TCP/IP, UDP, ATM, synchronous optical networks (SONET), wireless protocols, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), etc." Vasseur '879, 7:61-8:6.<br><br>"The memory 240 comprises a plurality of storage locations that are addressable by the processor 220 and the network interfaces 210 for storing software programs and data structures associated with the present invention. The processor 220 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures. A router operating system 242 (e.g., the Internetworking Operating System, or IOS™, of Cisco Systems, Inc.), portions of which is typically resident in memory 240 and executed by the processor, functionally organizes the router by, inter alia, invoking network operations in support of software processes and/or services executing on the router. These software processes and/or services may comprise routing services 247, Traffic Engineering (TE) services 244, and RSVP services 249. It will be apparent to those skilled in the art that other processor and memory means, including various computer-readable media, may be used to store and execute program instructions pertaining to the inventive technique described herein." Vasseur '879, 8:7-26.<br><br>"Routing services 247 contain computer executable instructions executed by processor 220 to perform functions provided by one or more routing protocols, such as IGP (e.g., OSPF and IS-IS), IP, BGP, etc. These functions may be configured to manage a forwarding information database (not shown) containing, e.g., data used to make forwarding decisions. Routing services 247 may also perform functions related to virtual routing protocols, such as maintaining VRF instances (not shown) as will be understood by those skilled in the art." Vasseur '879, 8:27-36. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "RSVP services 249 contain computer executable instructions for implementing RSVP and processing RSVP messages in accordance with the present invention. RSVP is described in RFC 2205, entitled Resource ReSerVation Protocol (RSVP), and in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels, both as incorporated above." Vasseur '879, 8:37-42.<br><br>"TE services 244 contain computer executable instructions for operating TE functions in accordance with the present invention. Examples of Traffic Engineering are described in RFC 3209, RFC 3784, and RFC 3630 as incorporated above, and in RFC 3473, entitled, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions dated January 2003, which is hereby incorporated by reference in its entirety. A TE database (TED, not shown) may be illustratively resident in memory 240 and used to store TE information provided by the routing protocols, such as IGP, BGP, and/or RSVP (with TE extensions, e.g., as described herein), including, inter alia, path quality information as described herein. The TED may be illustratively maintained and managed by TE services 244." Vasseur '879, 8:43-57.<br><br>"Changes in the network topology may be communicated among routers 200 using a link-state protocol, such as the conventional OSPF and IS-IS protocols. Suppose, for example, that a communication link fails or a cost value associated with a network node changes. Once the change in the network's state is detected by one of the routers, that router may flood an IGP Advertisement communicating the change to the other routers in the network. In this manner, each of the routers eventually "converges" to an identical view of the network topology." Vasseur '879, 8:58-67.<br><br>"In one embodiment, the routers described herein are IP routers that implement Multi-Protocol Label Switching (MPLS) and operate as label switched routers (LSRs). In one simple MPLS scenario, at an ingress to a network, a label is assigned to each incoming packet based on its forwarding equivalence class before forwarding the packet to a next-hop router. At each router, a forwarding selection and a new substitute label are determined by using the label found in the incoming packet as a reference to a label forwarding table that includes this information. At the network egress, a forwarding decision is made based on the incoming label |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | but optionally no label is included when the packet is sent on to the next hop. In some network configurations, one hop prior to the network egress, a penultimate hop popping (PHP) operation may be performed. Particularly, because the hop prior to the network egress (the penultimate hop) is attached to the network egress, the label is no longer needed to assure that the traffic follows a particular path to the network egress. As such, the PHP-enabled device "pops" the labels from the traffic before forwarding the traffic to the network egress, e.g., using conventional or native (IP) routing, thereby alleviating the task of removing the labels at the network egress." Vasseur '879, 9:1-23.<br><br>"The paths taken by packets that traverse the network in this manner are referred to as label switched paths (LSPs) or Traffic Engineering (TE)-LSPs. An example TE-LSP is shown as the thick line and arrow (T1) between a head-end node (router A) and a tailend node (router E) in FIG. 1. Establishment of a TE-LSP requires computation of a path, signaling along the path, and modification of forwarding tables along the path. MPLS TE establishes LSPs that have guaranteed bandwidth under certain conditions. Illustratively, the TE-LSPs may be signaled through the use of the RSVP protocol (with Traffic Engineering extensions), and in particular, RSVP TE signaling messages. Notably, when incorporating the use of PCEs (described below), the path computation request (and response) between PCC and PCE can be exchanged in accordance with a protocol specified in Vasseur, et al., Path Computation Element (PCE) Communication Protocol (PCEP)—Version 1—<draft-vasseur-pce-pcep-02.txt>, Internet Draft, September 2005, the contents of which are hereby incorporated by reference in its entirety. It should be understood that the use of RSVP or PCEP serves only as an example, and that other communication protocols may be used in accordance with the present invention." Vasseur '879, 9:24-45.<br><br>"In accordance with RSVP, to request a data flow (TE-LSP) between a sender and a receiver, the sender may send an RSVP path request (Path) message downstream to the receiver along a path (e.g., a unicast route) to identify the sender and indicate e.g., bandwidth needed to accommodate the data flow, along with other attributes of the TE-LSP. The Path message may contain various information about the data flow including, e.g., traffic characteristics of the data flow. Also in accordance with the RSVP, a receiver establishes the TE-LSP between the sender and receiver by responding to the sender's Path message with a reservation request |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | (Resv) message. The reservation request message travels upstream hop-by-hop along the flow from the receiver to the sender. The reservation request message contains information that is used by intermediate nodes along the flow to reserve resources for the data flow between the sender and the receiver, to confirm the attributes of the TE-LSP, and provide a TE-LSP label. If an intermediate node in the path between the sender and receiver acquires a Path message or Resv message for a new or established reservation (TE-LSP) and encounters an error (e.g., insufficient resources, failed network element, etc.), the intermediate node generates and forwards a path or reservation error (PathErr or ResvErr, hereinafter Error) message to the sender or receiver, respectively." Vasseur '879, 9:46-10:2.<br><br>"FIG. 3 is a schematic block diagram of portions of a signaling message 300 (e.g., RSVP message, such as Path, Resv or Error) that may be advantageously used with the present invention. Message 300 contains, inter alia, a common header 310 and one or more signaling protocol specific objects 320, such as an LSP-ATTRIBUTE object 330. The common header 310 may comprise a source address 312 and destination address 314, denoting the origination and requested termination of the message 300. Protocol specific objects 320 contain objects necessary for each type of message 300 (e.g., Path, Resv, Error, etc.). For instance, a Path message may have a sender template object, Tspec object, Previous-hop object, etc. The LSP-ATTRIBUTE object 330, for instance, may be used to signal attributes and/or information regarding an LSP (tunnel). To communicate this information, LSP-ATTRIBUTE object 330 (as well as specific objects 320) may include various type/length/value (TLV) encoding formats and/or flags, as will be understood by those skilled in the art. An example of an LSP-ATTRIBUTE object is further described in Farrel, et al., Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE <draft-ietf-mpls-rsvpte-attributes-05.txt>, Internet Draft, May 2005, which is hereby incorporated by reference as though fully set forth herein. A Resv message, on the other hand, may have specific objects 320 for a label object, session object, filter spec object, etc., in addition to the LSP-ATTRIBUTE object 330. Error messages 300 (e.g., PathErr or ResvErr), may also have specific objects 320, such as for defining the type of error, etc." Vasseur '879, 10:3-31. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "It should be noted that in accordance with RSVP signaling, the state of the TE-LSP is refreshed on a timed interval, e.g., every thirty seconds, in which RSVP Path and Resv messages are exchanged. This timed interval is configurable by a system administrator. Moreover, various methods understood by those skilled in the art may be utilized to protect against route record objects (RROs) contained in signaling messages for a TE-LSP in the event security/privacy is desired. Such RRO filtering prevents a head-end node of the TE-LSP from learning of the nodes along the TE-LSP, i.e., nodes within the provider network." Vasseur '879, 10:4-42.<br><br>"Although the illustrative embodiment described herein is directed to MPLS, it should also be noted that the present invention may advantageously apply to Generalized MPLS (GMPLS), which pertains not only to packet and cell-based networks, but also to Time Division Multiplexed (TDM) and optical networks. GMPLS is well known and described in RFC 3945, entitled Generalized Multi-Protocol Label Switching (GMPLS) Architecture, dated October 2004, and RFC 3946, entitled Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control, dated October 2004, the contents of both of which are hereby incorporated by reference in their entirety." Vasseur '879, 10:43-55.<br><br>"To obviate delays associated with updating routing tables when attempting to avoid a failed network element (i.e., during convergence), some networks have employed MPLS TE Fast Reroute (FRR). MPLS Fast Reroute is a technique that may be used to quickly divert ("Fast Reroute") traffic around failed network elements in a TE-LSP. MPLS Fast Reroute is further described, for example, by Fast Reroute Extensions to RSVP-TE for LSP Tunnels, as incorporated by reference above. According to the technique, one or more network elements (e.g. links or nodes) in a network are protected by backup tunnels following an alternate path. If a failure occurs on a protected link or node, TE-LSPs (and consequently the traffic that they carry) are locally diverted onto an appropriate alternate path (e.g., a "backup tunnel") by the node immediately upstream from the failure. The backup tunnel acts as a Fast Reroute path for the primary TE-LSP and obviates delays associated with other measures, such as tearing down the primary TE-LSP after having gracefully diverted the TE-LSPs affected by the failure, should an alternate path around the failed network element exist. In the event of a |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | failure of a protected element the head-end node of the backup tunnel (or a "point of local repair," PLR node) may quickly begin diverting traffic over the backup tunnel with minimal disruption to traffic flow. Those skilled in the art will understand that MPLS Fast Reroute is one example of link or node failure protection, and that other known correction mechanisms may be used in accordance with the present invention. As mentioned above, however, the head-end node of the Fast Rerouted primary tunnel may attempt to reestablish the primary tunnel in response to learning of the protected element failure, particularly in the case where the backup tunnel is a zero-bandwidth (0-BW) tunnel. The attempt to reestablish the primary tunnel has conventionally been a systematic response to Fast Rerouting (diverting) of the primary tunnel, regardless of the path quality of the backup tunnel." Vasseur '879, 10:56-11:23.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 11:24-39.<br><br>"In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as RSVP TE signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or TLV encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object 330 of the RSVP messages 300 to convey the path quality notification." Vasseur '879, 11:40-49. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 4 is a schematic block diagram illustrating the format of an extension object (TLV) 400, such as a Fast Reroute Feedback object, that may be advantageously used with the present invention. The extension object (or sub-object) 400 is illustratively embodied as a TLV contained in an LSP-ATTRIBUTE object 330 of an RSVP message 300 and is extended to carry backup (and/or primary) tunnel path quality information. To that end, the "Feedback" object 400 is organized to include a Type field 405 containing a predetermined type value signifying the specific content of the object 400. The Length field 410 is a variable length value. The TLV encoded format may also comprise one or more non-ordered sub-TLVs 450 carried within the TLV "payload" (e.g. Value field 415), each having a Type field 455, Length field 460, and Value field 465. The fields of the TLV 400 and sub-TLV(s) 450 are used in a variety of manners, including as described herein, according to the present invention." Vasseur '879, 11:59-12:8.<br><br>"In accordance with one aspect of the present invention, the head-end node (e.g., router A) requests the establishment of the primary tunnel (e.g., T1), such as a TE-LSP. Notably, the head-end node may be a head-end node for multiple primary tunnels, as will be understood by those skilled in the art. Along with the primary tunnel establishment, the head-end node may also request Fast Reroute protection of one or more network elements (e.g., all intermediate network elements) at a PLR node (e.g., router B as shown). Note that each intermediate node along the primary tunnel may act as a PLR node, and that router B is shown merely for simplicity. Illustratively, the Fast Reroute protection may be embodied as one or more zero-bandwidth (0-BW) backup tunnels at the PLR node (e.g., BT1). Those skilled in the art will also understand that the PLR node may protect more than one primary tunnel originating at more than one corresponding head-end node (not shown)." Vasseur '879, 12:9-25.<br><br>"The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. For example, various connectivity verification protocols, such as, e.g., Bidirectional Forwarding Detection (BFD), IGP "Hello" packets, BGP KEEPALIVE messages, etc., may be used to detect a failure of a network element, as will be understood by those skilled in the art. Furthermore, other lower-layer failure detection mechanisms (e.g. optical or SONET/SDH alarms) may be used to detect a network element failure. Once failure is detected, the PLR node diverts the |

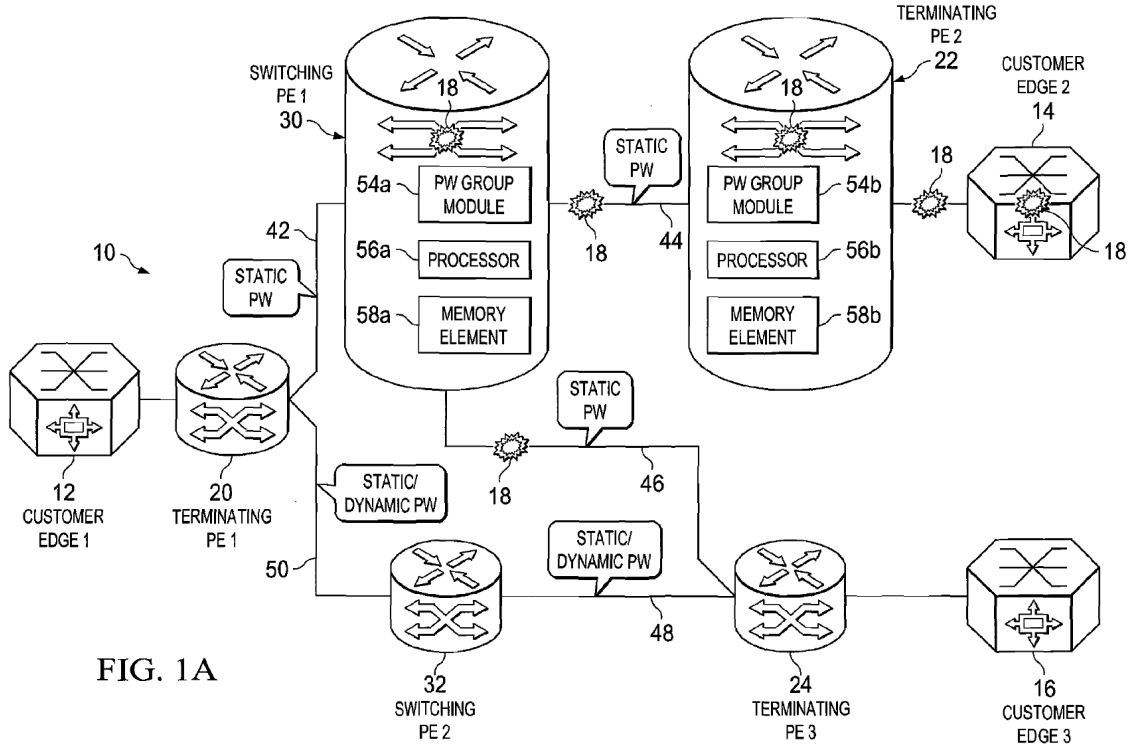
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | traffic onto the backup tunnel, and may send an error message (e.g., an RSVP PathErr 300, such as a conventional "tunnel locally repaired" message) to the head end node indicating the "Fast Rerouting" of the primary tunnel. FIG. 5 is a schematic block diagram of the computer network 100 in FIG. 1 showing Fast Reroute protection of the primary tunnel T1 (e.g., in response to a protected network element, router C, failure, indicated with an overlaid "X") using a backup tunnel BT1 in accordance with the present invention. Traffic originally received at the PLR node (router B) over the primary tunnel is now diverted over the backup tunnel to a remerge point (router D) of the primary tunnel, as will be understood by those skilled in the art." Vasseur '879, 12:42-65.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, each PLR node may collect metrics/statistics of the primary and/or backup tunnels. For instance, example metrics may comprise, inter alia, packet drops, path cost, jitter, delay, bandwidth, etc. The PLR node may collect the metrics through traffic monitoring, probes, independent calculations, and/or through cooperation with protected nodes of the primary tunnel (primary nodes) and nodes of the backup tunnel (backup nodes), e.g., transmitting path quality notifications. Once the primary tunnel is Fast Rerouted (i.e., diverted after failure of a protected network element), the PLR node continues to collect metrics of the backup tunnel. (Alternatively, metrics of the backup tunnel may be collected only after Fast Reroute, and not prior to Fast Reroute.)" Vasseur '879, 12:66-13:12.<br><br>"For example, a path cost increase from the primary tunnel to the backup tunnel may be computed (and transmitted, below) by the PLR node prior to the failure (or during the failure while the primary TE-LSP is diverted onto the backup tunnel) using its own routing tables. The path cost increase may be calculated as a difference between the entire length (head-end node to tail-end node) of the primary and backup tunnels, or just the difference between the protected segment of the primary tunnel and the backup tunnel (PLR node to remerge point). Also, a jitter increase of the primary and backup tunnels, which may be generally described as a difference between inter-arrival of packets, may be monitored using various known techniques, such as, e.g., sending probe packets (probes) from the PLR node to the remerge point. For instance, probes may determine that packets arrive at the remerge point along the primary tunnel from the PLR node consistently, e.g., every 10 milliseconds (ms) (e.g., an |

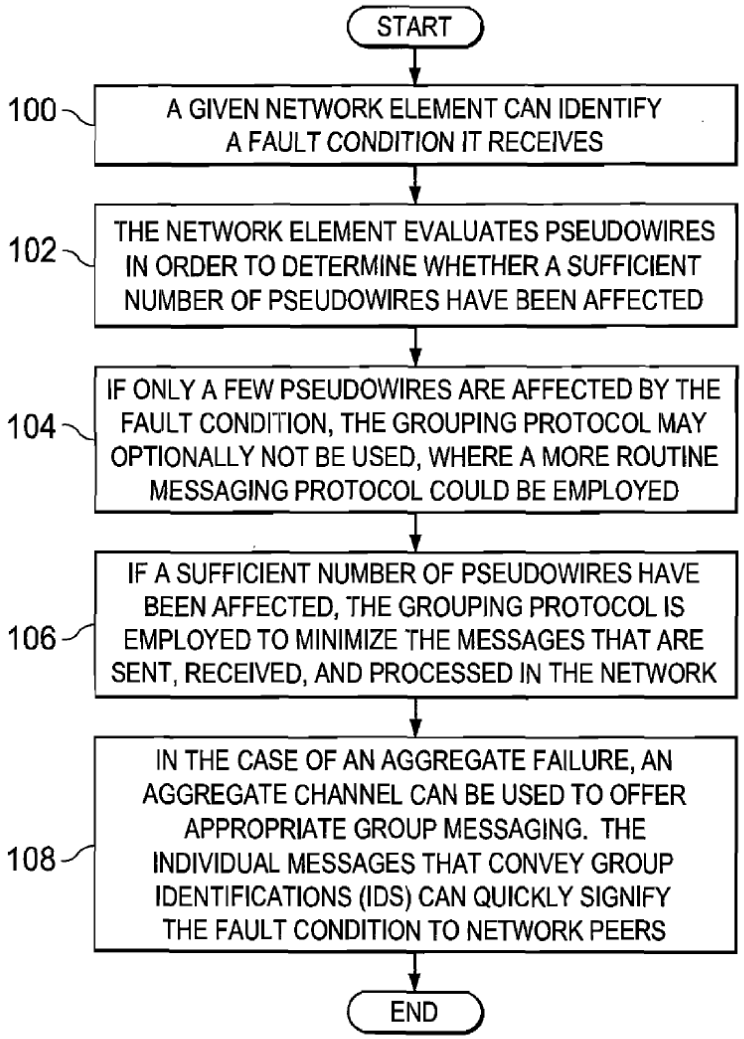
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | average value). After Fast Reroute, however, probes may determine that packets do not arrive at the remerge point along the backup tunnel from the PLR node consistently, e.g., one may arrive in 10 ms, another in 100 ms, another in 50 ms, etc. The non-constant rate of received packets (jitter) may be undesirable, e.g., in particular for voice over IP (VoIP) traffic, as will be understood by those skilled in the art." Vasseur '879, 13:13-36.<br><br>"As a further example, packet dropping may be monitored for the primary and backup tunnels prior to and after Fast Reroute (respectively). For instance, based on the tunnel label of the dropped packet, primary nodes and/or backup nodes may be able to distinguish which tunnel corresponds to the dropped packets. Each of the primary and/or backup nodes collect packet drop data, and periodically inform the PLR node of the number of dropped packets (e.g., though a corresponding Feedback object 400). In the case of a backup node, the PLR node receiving the notification may interpret the association of the backup tunnel label and the primary tunnel label to reference an appropriate primary tunnel. Those skilled in the art will understand that the above path quality metrics are merely examples, and that any other metrics/statistics useful for determining path quality of the backup tunnel may be used in accordance with the present invention (e.g., delay, bandwidth, etc.). Further, the path quality information may be measured and compared in a variety of manners, such as, e.g., as a difference between primary and backup tunnels before and after Fast Reroute, or simply the difference between the backup tunnel before and after Fast Reroute, etc." Vasseur '879, 13:37-58.<br><br>"Also after the primary tunnel is Fast Rerouted, the PLR node may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. For instance, the novel Feedback object 400 may include one or more sub-TLVs 450 corresponding to metrics/statistics, as described above. Notably, the path quality information pertaining to a particular metric/statistic may be transmitted as total values for interpretation by the head-end node (e.g., to determine the difference), or as PLR-node-computed differences (e.g., between the primary and backup tunnels, or before and after Fast Reroute). For instance, if the delay of the primary tunnel (along the protected segment) prior to Fast Reroute were 2 ms, and after Fast Reroute the delay of the backup tunnel were 5 ms, the notification may be configured to |

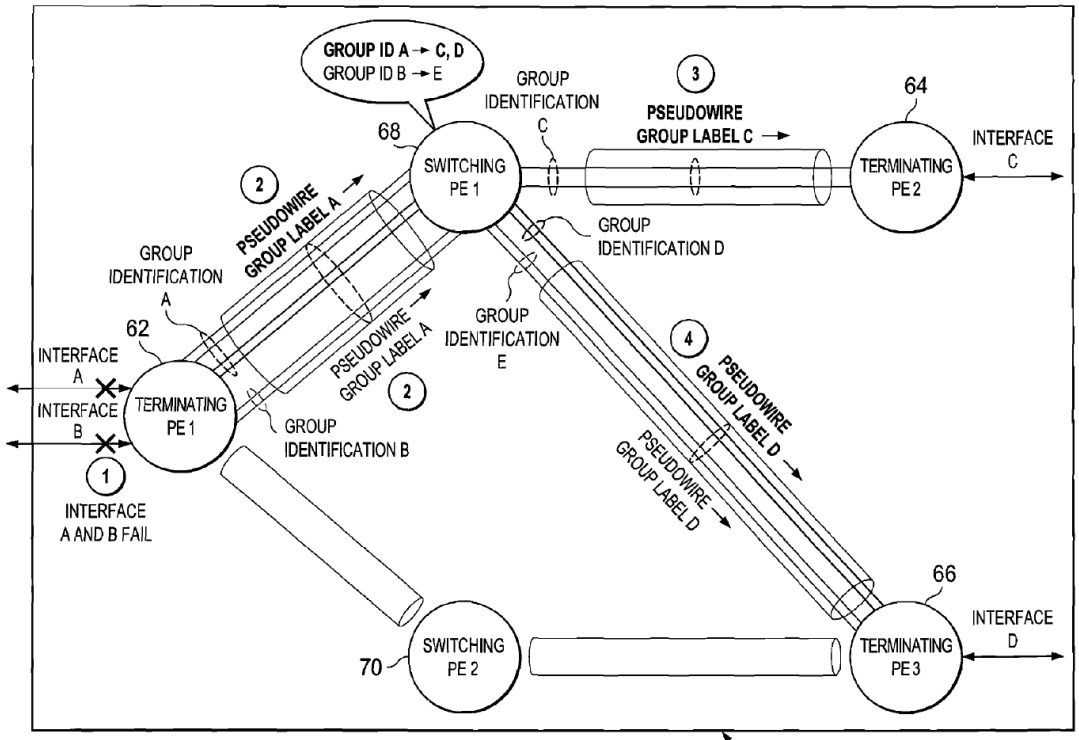
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | include both values 2 ms and 5 ms, or instead simply the difference, i.e., an increase of 3 ms." Vasseur '879, 13:59-14:8.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message 300 (PathErr), the head-end node may wait for at least one path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. Also, any number of metrics may be used in the determination, e.g., as configured by a system administrator. For example, using the metrics described above, a head-end node may be configured to reestablish the primary tunnel in response to i) a certain number of packet drops, ii) a percent increase in packet drops, iii) a number of packet drops and a percent increase in path cost, iv) a percent increase in path cost and a percent increase in jitter, etc. Those skilled in the art will understand that these are merely examples of possible path quality comparisons and reestablishment determinations, and that any comparisons to any metrics at any configurable changes may be used in accordance with the present invention." Vasseur '879, 14:41-59.<br><br>"If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. FIG. 6 is a schematic block diagram of the computer network 100 in FIG. 5 showing an unacceptable backup tunnel path quality (dotted line and arrow) and resultant reestablishment of the primary tunnel in accordance with the present invention. Those skilled in the art will understand that the reestablished primary tunnel may traverse one or more primary nodes (not shown), and that it may be computed specifically to avoid the failed network element and any network elements of the unacceptable backup tunnel. Those skilled in the art will also understand that the attempt to reestablish the primary tunnel may not be able to find an acceptable path, in which case the head-end node may continue to use the unacceptable backup tunnel or other unacceptable rerouted path." Vasseur '879, 14:60-15:9. |

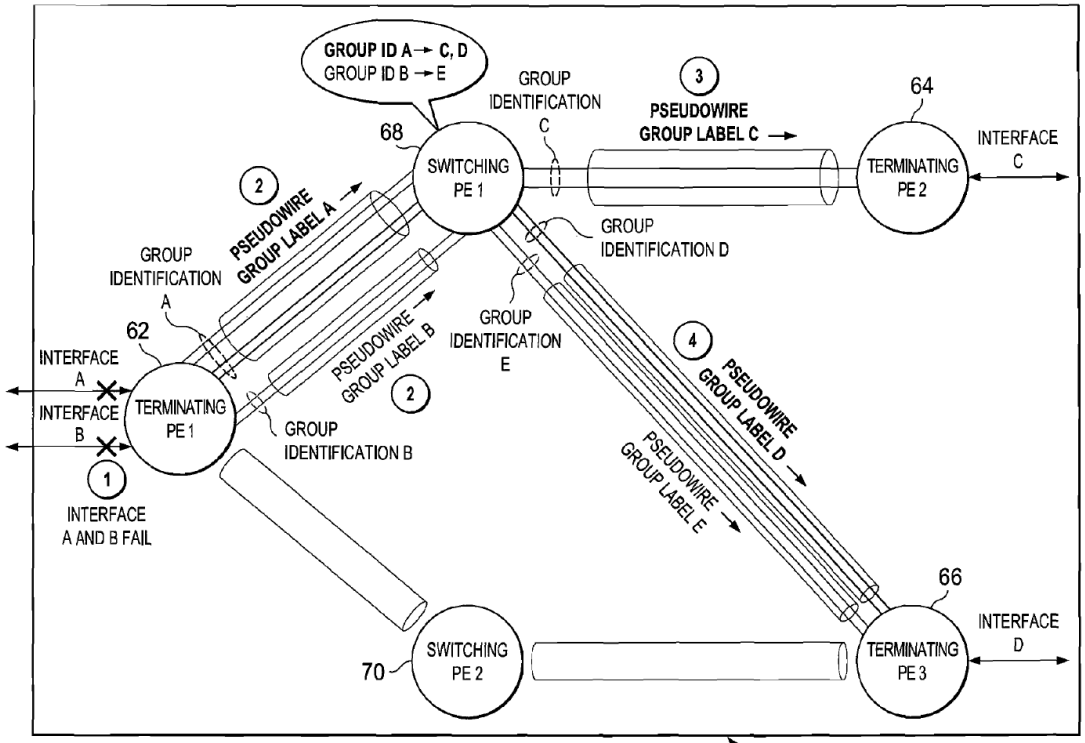
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention. The procedure 700 starts at step 705, and continues to step 710, where a head-end node (e.g., router A) establishes a primary tunnel (e.g., T1) to a destination tail-end node (e.g., router E). During or after establishment, the head-end node requests protection of the primary tunnel at step 715, and in response, PLR nodes along the primary tunnel (e.g., router B) create backup tunnels (e.g., BT1) to protect the primary tunnel in step 720. (Those skilled in the art will understand that backup tunnels around the protected network elements may already exist at the PLR node, and that "creating backup tunnels" in step 720 implies an association with pre-existing backup tunnels.) As mentioned above, these backup tunnels may illustratively be embodied as 0-BW backup tunnels. In accordance with the present invention, the head-end node may additionally request backup tunnel path quality notification from the PLR nodes in step 725, such as, e.g., through the use of empty corresponding Feedback objects in RSVP (Path) messages 300, as described above." Vasseur '879, 15:37-58.<br><br>"The procedure 700 continues to FIG. 7B (step "A"), where in step 750 the PLR node detecting the failure diverts ("Fast Reroutes") the primary tunnel traffic to the backup tunnel and sends an error message (e.g., an RSVP (Error) message 300) to the head end node, e.g., a "tunnel locally repaired" message. The detecting PLR node continues to collect path quality information for the backup tunnel in step 755 and at step 760 determines whether to send the path quality notification to the head-end node. For example, as mentioned above, the PLR node may be configured to continually send notifications, or periodically, or in response to a configurable change in path quality, etc. Also as mentioned above, the PLR node may be configured to send either the actual path quality information or the change (difference) in path quality. (As further mentioned above, metrics of the backup tunnel may alternatively be collected only after Fast Reroute, and not prior to Fast Reroute.)." Vasseur '879, 16:4-20.<br><br>"If the PLR node decides to send the notification in step 760, then the head-end node determines whether to reestablish the primary tunnel based on the backup tunnel path quality notification in step 765, e.g., based on one or more configurable thresholds, percentages, etc., as described above. If the backup tunnel is currently maintaining an acceptable quality for the |

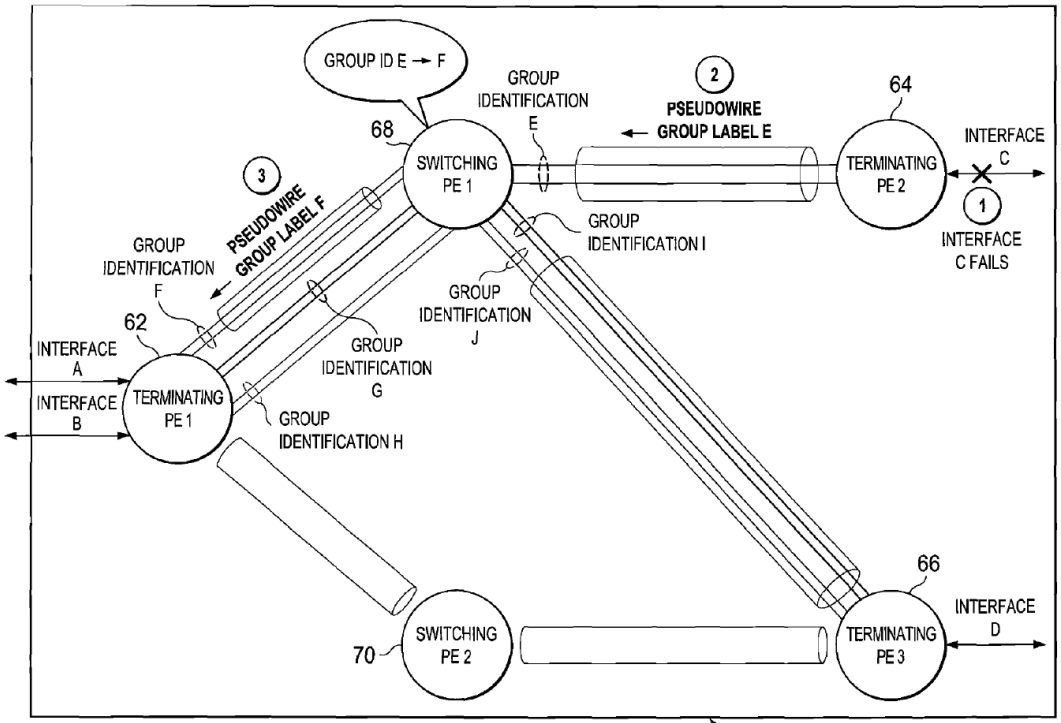
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | traffic flow in step 770, the head-end node may continue to utilize the backup tunnel, and the PLR node continues to collect path quality information in step 755 to detect any change in quality. Otherwise, if the backup tunnel quality is not acceptable in step 770, the head-end node may attempt to reestablish the primary tunnel in step 775. Notably, as mentioned above, if the PLR node has not sent any notification (step 760) within a configurable period of time in step 780, e.g., due to a backup tunnel failure, over-congestion, etc., then the head-end node may also attempt to reestablish the primary tunnel in step 775 accordingly. Moreover, as described above, in the event more than one primary tunnel is Fast Rerouted for the head-end node, various techniques to reestablish one or more of the primary tunnels may be used (e.g., as many tunnels as necessary, a configurable subset of tunnels, all tunnels, the congested tunnels, etc.). The procedure 700 ends in step 785." Vasseur '879, 16:21-43.<br><br>"While there has been shown and described an illustrative embodiment that dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the present invention. For example, the invention has been shown and described herein using "Fast Reroute," e.g., MPLS TE Fast Reroute (FRR). However, the invention in its broader sense is not so limited, and may, in fact, be used with other network element protection and failure correction mechanisms as will be understood by those skilled in the art. Moreover, while the above description describes performing the technique at the head-end node and PLR node, the invention may also be advantageously used with PCEs. In addition, while one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 16:63-17:13.<br><br>**Rustogi discloses**:<br>"An example method includes identifying a fault condition in a network, and evaluating pseudowires affected by the fault condition in order to make a determination as to whether an aggregate failure occurred in the network for a group of pseudowires. The method also includes communicating a group message indicating that the group of pseudowires is associated with the fault condition. The group message includes a group identification (ID), |

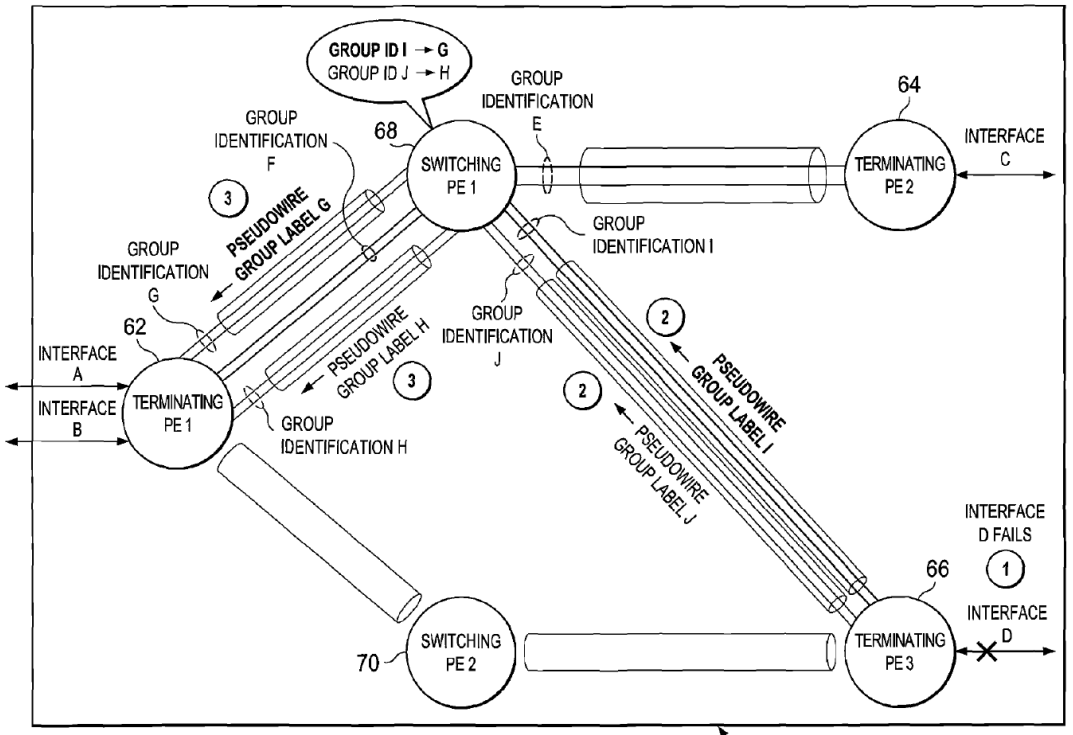
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | which identifies the group of pseudowires, and the group message includes a pseudowire group label identifying an in-band aggregate channel. More specifically, the pseudowire group label can be applicable to static pseudowires. In more detailed embodiments, the group ID identifies the group of pseudowires that are associated with an attachment circuit, a label switched path, or a port. Internal mappings can be maintained such that a plurality of pseudowires is mapped to individual interfaces of network elements in the network." Rustogi, Abstract.<br>  <br><br>FIG. 1A<br>Rustogi, FIG. 1A. |

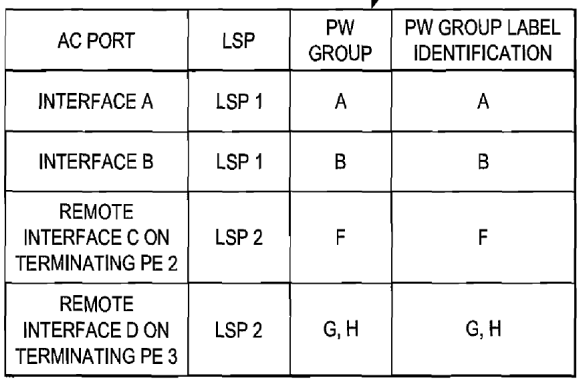
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 1B<br>Rustogi, FIG. 1B. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Rustogi, FIG. 2. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Rustogi, FIG. 3. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Rustogi, FIG. 4. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Rustogi, FIG. 5. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | 74<br>  <br>FIG. 6<br>Rustogi, FIG. 6.<br><br>"The field of communications has become increasingly important in today's society. In particular, the ability to quickly and to effectively provision connections presents a significant challenge to component manufacturers, system designers, and network operators. Multiprotocol Label Switching (MPLS) is a mechanism in telecommunications networks that carries data from one network node to the next. Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be emulated over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDUs) and transmitting them over pseudowires. Protocols exist for establishing and maintaining the pseudowires. Certain issues have arisen in pseudowire scenarios, where faults are detected in the network." Rustogi, ¶ [0002].<br><br>"FIG. 1A is a simplified block diagram of a communication system for providing pseudowire group labels in a network environment in accordance with one embodiment of the present disclosure." Rustogi, ¶ [0004]. |

| AC PORT | LSP | PW GROUP | PW GROUP LABEL IDENTIFICATION |
|---|---|---|---|
| INTERFACE A | LSP 1 | A | A |
| INTERFACE B | LSP 1 | B | B |
| REMOTE INTERFACE C ON TERMINATING PE 2 | LSP 2 | F | F |
| REMOTE INTERFACE D ON TERMINATING PE 3 | LSP 2 | G, H | G, H |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 1B is a simplified flowchart depicting one possible, generic operational flow associated with the communication system." Rustogi, ¶ [0005].<br><br>"FIG. 2 is a simplified block diagram of an example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0006].<br><br>"FIG. 3 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0007].<br><br>"FIG. 4 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0008].<br><br>"FIG. 5 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0009].<br><br>"FIG. 6 is a simplified table of an example set of pseudowire group provisioning parameters in accordance with one embodiment." Rustogi, ¶ [0010].<br><br>"FIG. 1A is a simplified block diagram of a communication system 10 for providing pseudowire group labels in accordance with one example of the present disclosure. FIG. 1A includes a customer edge 1 (CE1) 12, a CE2 14, and a CE3 16, where a number of faults 18 are shown as propagating in the network. Typically, when an error or a failure occurs in the network (e.g., an interface failure, a pulled cable, a switch failure, hardware/software failures generally, etc.), messages are sent to various network devices in order to inform them of these fault conditions. Faults 18 of FIG. 1A are indicative of such messages, where the underlying fault condition (being signaled by the messages) can occur virtually anywhere in a network (e.g., in a customer edge, in provider equipment, etc.). FIG. 1A also includes a terminating provider equipment 1 (TPE1) 20, a TPE2 22, a TPE3 24, a switching provider edge 1 (SPE1) 30, and a SPE2 32. In one particular example implementation, the TPEs and SPEs of FIG. 1A are switches that are configured to exchange data in a network environment." Rustogi, ¶ [0012]. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "SPE1 30 may include a pseudowire (PW) group module 54 a, a processor 56 a, and a memory element 58 a. In a similar fashion, TPE2 22 may include a pseudowire group module 54 b, a processor 56 b, and a memory element 58 b. FIG. 1A also includes a number of static pseudowires 42, 44, and 46. A set of static/dynamic pseudowires 48, 50 is also provided. Note that the group labeling protocol discussed herein can be executed between individual SPEs, TPEs, or between any combinations of these elements." Rustogi, ¶ [0013].<br><br>"In one particular arrangement, communication system 10 is provided in conjunction with a Layer-2 virtual private networks (L2VPN)/operation, administration, and maintenance (OAM) L2VPN/OAM framework. The OAM framework is intended to provide OAM layering across L2VPN services, pseudowires, and packet switched network (PSN) tunnels. Communication system 10 may include any suitable networking protocol or arrangement that provides a communicative platform for communication system 10. Thus, communication system 10 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Communication system 10 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs." Rustogi, ¶ [0014].<br><br>"Failure detection and failure notification for static pseudowires is inadequate, where sluggish signaling can result in poor scalability for failover performance. Typically, static pseudowires are manually configured at respective endpoints, where control channels are absent for providing group level signaling messages. Aggregate channels are significant tools for providing suitable scalability in the network, but no such aggregate channel exists for static pseudowires. For dynamic pseudowires, such an aggregate channel may be present in the form of a label distribution protocol (LDP) directed session. However, no such protocol exists for static pseudowire configurations such that an in-band aggregate channel would be available for static pseudowires." Rustogi, ¶ [0016]. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Communication system 10 can address the aforementioned issues (and others) by offering a pseudowire group label that can represent an aggregate channel for groups of static pseudowires. The aggregate channel of communication system 10 can allow for improved scalability of failover performance. In accordance with one potential configuration of communication system 10, a pseudowire group label is representative of a group of static pseudowires transported over a label switched path (LSP). The pseudowire group label can identify the aggregate channel, which captures the hierarchy relevant to OAM mechanisms. Additionally, the groups represented by the group identification (ID) can be mutually exclusive, where a pseudowire is part of only one group. In other embodiments, a pseudowire can be part of multiple groups, or be configured in any other suitable manner based on particular network arrangements." Rustogi, ¶ [0017].<br><br>"During operations, and with brief reference to FIG. 1B, a given network element can identify a fault condition it receives (at step 100) and, subsequently, evaluate pseudowires in order to determine whether a sufficient number of pseudowires have been affected. This is reflected by step 102. If only a few pseudowires are affected by the fault condition, the grouping protocol outlined herein may have only nominal value, where there could be an option to simply communicate the fault condition in a more routine manner, as outlined in step 104. However, if a sufficient number of pseudowires have been affected, the grouping protocol outlined herein can be employed to minimize the messages that are sent, received, and processed in the network. This is reflected as step 106. Note that the determination (as to whether a sufficient number of pseudowires have been impacted by the fault condition) can involve accessing internal tables such that a quick mapping can occur to determine if an aggregate failure has occurred. As used herein, the term 'aggregate failure' simply connotes that a sufficient number of pseudowires have experienced the fault condition such that an aggregate channel can be employed to offer appropriate group messaging. For the aggregate failure condition, the individual messages that convey Group identifications (IDs) can quickly signify the fault condition to network peers, as shown in step 108." Rustogi, ¶ [0018]. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "In specific regards to OAM mechanisms, OAM messages typically result from common failures in the network. These fault conditions can be aggregated such that they are signaled as a single message, which could represent a group of failed pseudowires (as opposed to sending individual messages for each failed pseudowire). Hence, a single message could be sent to represent all the relevant OAM messages propagating in communication system 10. The group label that propagates in communication system 10 provides an architecture with a significant level of aggregation for failed pseudowires (i.e., pseudowires being affected by a given fault condition), particularly for OAM messaging. Moreover, the in-band aggregate channel of communication system 10 is based (at least in part) on the evolving trends of OAM mechanisms, which are required to be fast, responsive, and capable of being implemented in hardware or software. Additionally, in-band OAM protocols are a better measure of the path availability." Rustogi, ¶ [0020].<br><br>"In operation of one example implementation, a group label can represent the tuple <attachment circuit (AC) port level grouping, LSP>. This could signify that all pseudowires on an AC port (sought for aggregation) traverse a given LSP. Multiple pseudowire groups can exist within an LSP. Similarly, pseudowires on the same AC port (that traverse a different LSP) can use a different pseudowire group label. Alternatively, an administrator may seek to employ a one-to-one mapping between an LSP and a group label. If that were the case, then only one pseudowire group would exist within an LSP. In scenarios where there is no LSP label in the packet (e.g., due to penultimate hop popping), the pseudowire group label can provide the hierarchy that is appropriate." Rustogi, ¶ [0021].<br><br>"In one particular example, the group level pseudowire OAM message can be sent with the following label stack: Explicit/Implicit LSP Label+pseudowire group Label+GAL+ACH+pseudowire OAM with grouping TLV (where GAL=Generic Associated Channel Label, ACH=Associated Channel Header, TLV=Type-Length-Value). If there are multiple LSPs, then one group label can be provisioned for each LSP (for each pseudowire group), where per group messages can be sent on each LSP. The group label does not necessarily have a one-to-one mapping to the grouping of pseudowires implied by the Group ID in the grouping TLV. Note also that the group-based aggregate channel is applicable to |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | static pseudowires, as well as for dynamic pseudowires in certain applications." Rustogi, ¶ [0022].<br><br>"As discussed herein, the aggregate channel of communication system 10 can be configured in various ways. For example, and with regards to a first option, a separate label may simply be used to identify a pseudowire group within an LSP. The association of an OAM message and a pseudowire group is straightforward. There could potentially be multiple pseudowire group labels per LSP. As a second option, one group label can be used to identify a common pseudowire group channel on the LSP. In this implementation, one pseudowire group label is provided per LSP. The OAM message association to a pseudowire group is not as simple as the first option. As a third option, one pseudowire is simply designated to convey grouping information (e.g., without using a group label). In this case, there is no need for a pseudowire group label. Again, the OAM message association to a pseudowire group is not as simple as the first option." Rustogi, ¶ [0023].<br><br>"Any combination of formatting (for the Group ID and the pseudowire group label) can be used in the group message to be communicated in the network. In one example, only one of these elements is communicated when an aggregate fault condition is detected, or these elements can be combined into a single unique identifier. In the most generic example, a group message would at least include the Group ID (identifying the pseudowires affected by the fault) and a pseudowire group label (identifying an aggregate channel for communicating the group message). In this generic sense, a pipe (the Group ID) within a pipe (the pseudowire group label) is being identified, where the group message is identifying both elements during an aggregate fault condition. Operational details of communication system 10 are described below with reference to FIGS. 2-6. Note that before turning to additional example flows and example embodiments of the present disclosure, a brief overview of the infrastructure of communication system 10 is provided." Rustogi, ¶ [0024].<br><br>"CE1 12, CE2 14, and CE3 16 represent devices, infrastructure, equipment, clients, or customers seeking to initiate a data session in communication system 10. These elements may can comprise a digital subscriber line access multiplexer (DSLAM), a router, a personal computer, a server, a switch, and/or other devices associated with data propagation. Further, |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | these elements may sit behind, or in front of, one or more of these identified devices. The term 'CE' may be inclusive of the devices identified above (e.g., a DSLAM, a switch, etc.), as well as devices used to initiate a communication, such as a console, a proprietary endpoint, a telephone, a cellular telephone, a bridge, a computer, a personal digital assistant (PDA), a laptop or an electronic notebook, or any other device, component, element, or object capable of initiating voice, audio, media, or data exchanges within communication system 10. The customer element may also include any device that seeks to initiate a communication on behalf of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating a voice, a video, text, or a data exchange within communication system 10. Data, as used herein in this document, refers to any type of video, numeric, voice, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another." Rustogi, ¶ [0025].<br><br>"SPE1 30, SPE2 32, TPE1 20, TPE2 22, and TPE3 24 are network elements that facilitate communications in two directions in a network environment. In one particular example, each of these network elements is a switch configured to exchange data over static and/or dynamic pseudowire links. Further, the traffic exchanged between these components may be directed over an MPLS transport in certain embodiments. As used herein in this Specification, the term 'network element' is meant to encompass switches, routers, bridges, gateways, servers, processors, loadbalancers, firewalls, or any other suitable device, component, element, or object operable to exchange or process information in a network environment. Moreover, these network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information. Along similar design alternatives, any of the internal modules and components of these network elements may be combined in various possible configurations." Rustogi, ¶ [0029]. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "Turning to FIG. 2, FIG. 2 is a simplified block diagram of an example system 60 for providing an example use case using per-label switched path (LSP) pseudowire group labels. FIG. 2 includes a TPE1 62, a TPE2 64, a TPE3 66, a SPE1 68, and a SPE2 70. Each pseudowire group is identified, where a group identification (ID) for Group A and Group B is depicted at TPE1 62. Similarly, Groups C, D, and E have Group IDs at SPE1 68. TPE2 64 and TPE3 66 can couple to interfaces C and D, respectively." Rustogi, ¶ [0031].<br><br>"In this particular example, interfaces A and B have failed. Note that there is a multitude of attachment circuits (e.g., 1000 attachment circuits) that are being transported over these interfaces A and B, where the attachment circuits are being tunneled into a corresponding number of pseudowires. For example, there could be 500 attachment circuits on interface A (implicating 500 pseudowires) and 500 attachment circuits on interface B, where the fault condition for the pseudowires should be signaled. In other flawed systems, an architecture would individually signal this fault condition for each pseudowire (e.g., via signaling between TPE1 62 and SPE1 68). Instead of sending 500 messages, a single message can be sent, where a single label (and Group ID) can be used to identify the pseudowires. In this case, the Group ID A is used to signal the fault condition for 300 pseudowires and for 200 pseudowires (i.e., the top two links connecting TPE1 62 and SPE1 68) using a single message (that includes Group Label A and Group ID A). Thus, the status for Group A is quickly communicated to SPE1 68. Similarly, Group ID B can be used to signal the status of the other 500 pseudowires to appropriately convey the status for Group B. More specifically, the message can include Group Label A and Group ID B. Note that all 1000 pseudowires have effectively been accounted for using these Group IDs A and B." Rustogi, ¶ [0032].<br><br>"FIG. 3 is a simplified block diagram of an example system 72 for providing another use case for pseudowire group labels. Note that the grouping mechanism outlined herein is not limited to pseudowires that propagate over LSPs. Certain pseudowires can propagate over an LSP and represent one group, where two ports can be provisioned for two different groups (e.g., Group A and Group B). Hence, FIG. 3 is depicting a use case using pseudowire group labels for <port, LSP>mapping. In a general sense, such a configuration is showing how pseudowire mechanics can be used to offer different group signaling, which may be based on various possible implementations. Thus, there is a group level construct corresponding to the group |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | labels that are created such that any OAM protocol can send the appropriate aggregate messages. In this particular example, the signaling for Group ID A, B, C, and D is similar to that of FIG. 2; however, the grouping mechanism has simply changed." Rustogi, ¶ [0035].<br><br>"FIG. 4 is a simplified block diagram of an example system 76 for providing another use case for pseudowire group labels. In this particular example, interface C fails (as shown at TPE2 64). Note that the same logical flow occurs in FIG. 4 in terms of the group signaling, as previously discussed. The group labels in two directions do not have to be the same, where the groupings for the messaging are not necessarily symmetrical. In this particular example, TPE2 64 sends a status for Group E with the corresponding group label (i.e., Group ID E for 300 pseudowires), where that message will have a Group Label E and a Group ID E. Hence, this particular signaling is indicative of 300 pseudowires failing in the network. SPE1 68 can send the status for Group F (where the Group ID F is associated with 300 pseudowires) to TPE1 62, where that message includes a Group Label F and a Group ID F." Rustogi, ¶ [0036].<br><br>"FIG. 5 is a simplified block diagram of an example system 80 for providing another use case for pseudowire group labels. In this particular example, interface D fails (as shown at TPE3 66), where all 700 pseudowires fail. In one implementation, TPE3 66 does not have a 700 pseudowire Group ID. Instead, the Group IDs can correspond to 200 and 500 pseudowires, when summed together account for the 700 pseudowires. In this particular example, TPE3 66 sends one message for Group I (representing 200 pseudowires) and another message for Group J (representing 500 pseudowires) to SPE1 68. In response, SPE1 68 sends a message for Group G (representing 200 pseudowires) and another message for Group H (representing 500 pseudowires). Again, the signaling being exchanged between these elements is minimal due to the effective grouping of pseudowires. SPE1 68 also sends a single message for Group I (associated with 200 pseudowires) and Group J (associated with 500 pseudowires) to TPE3 66, which is coupled to interface D. Group ID G is associated with 200 pseudowires, whereas Group ID H is representative of 500 pseudowires." Rustogi, ¶ [0037]. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "FIG. 6 is a simplified table 74 illustrating an example set of pseudowire group provisioning parameters for TPE1 62, where these particular provisioning parameters could be relevant to the configuration of FIG. 3. At least in one generic sense, FIG. 2 can reflect one approach for mapping a PW group label to a PW Group ID, while FIGS. 3-5 can reflect a second approach for such mappings, where table 74 is associated with that second approach." Rustogi, ¶ [0038].<br><br>"In particular, table 74 illustrates the mapping between SPE1 68 and TPE1 62. The first column represents the attachment circuit port (e.g., interface A, interface B, remote interface C on TPE2 64, and remote interface D on TPE3 66). Additionally, table 74 depicts a number of LSPs, a set of pseudowire grouping labels, and a set of pseudowire Group IDs. Note that the Group IDs are provided inside the pseudowire group labels in this example such that these two columns match in table 74. Additionally, note that table 74 is merely representing some of the possible characteristics in a single direction, where different constructs could be used in the reverse direction. Note that the provisioning as discussed herein can significantly reduce messaging such that these presented concepts offer increased scalability. This is due in part to the nominal processing that occurs in the network, in contrast to the processing required to evaluate a prolific amount of signaling messages associated with particular pseudowires. Additionally, the paradigm discussed herein can afford service providers an adequate amount of downtime after a failure has occurred in the network." Rustogi, ¶ [0039]. |
| 14[a] | a data structure comprising a plurality of transport entity descriptors; | The Reference discloses a data structure comprising a plurality of transport entity descriptors.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| 14[b] | an entity protection switch configured to switch between a working entity and a protection entity; and | The Reference discloses an entity protection switch configured to switch between a working entity and a protection entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |
| 14[c] | digital logic configured to select said working entity and said protection entity from said plurality of transport entity descriptors, comprising: | The Reference discloses digital logic configured to select said working entity and said protection entity from said plurality of transport entity descriptors, comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See infra* claim14[d]-[e]. |
| 14[d] | logic configured to determine a probability of concurrent failure of said working entity and said protection entity; | The Reference discloses logic configured to determine a probability of concurrent failure of said working entity and said protection entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| 14[e] | logic configured to determine an entity cost of said plurality of transport entity descriptors: and | The Reference discloses logic configured to determine an entity cost of said plurality of transport entity descriptors.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>Below are examples of such references.<br><br>**Kurose discloses**:<br>For example, Kurose discloses the well-known algorithm of calculating the least-cost between devices when forming a network path.<br><br>"The purpose of a routing algorithm is simple: given a set of routers, with links connecting the routers, a routing algorithm finds a 'good' path from source to destination. Typically, a 'good' path is one that has 'least cost.'" Kurose at 280.<br><br>"A link also has a value representing the 'cost' of sending a packet across the link. The cost may reflect the level of congestion on that link (for example, the current average delay for a packet across that link) or the physical distance traversed by that link (for example, a transoceanic link might have a higher cost than a short-haul terrestrial link). For our current purposes, we'll simply take the link costs as a given and won't worry about how they are determined." Kurose at 280. |

| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|

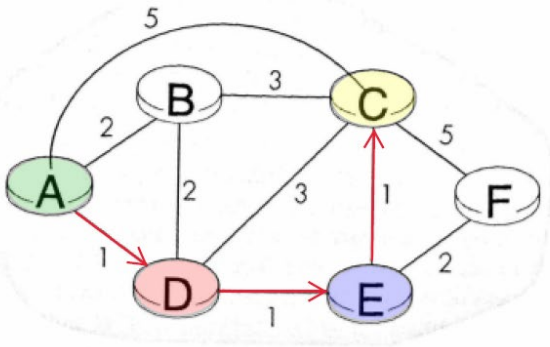



**Figure 4.4** ✦ Abstract model of a network

- the first link in the path is connected to the source
- the last link in the path is connected to the destination
- for all $i$, the $i$ and $i-1$st link in the path are connected to the same node
- for the **least-cost path,** the sum of the cost of the links on the path is the minimum over all possible paths between the source and destination. Note that if all link costs are the same, the least-cost path is also the **shortest path** (that is, the path crossing the smallest number of links between the source and the destination).

In Figure 4.4, for example, the least-cost path between nodes $A$ (source) and $C$ (destination) is along the path $ADEC$. (We will find it notationally easier to refer to the path in terms of the nodes on the path, rather than the links on the path.)

As a simple exercise, try finding the least-cost path from nodes $A$ to $F$, and reflect for a moment on how you calculated that path. If you are like most people, you found the path from $A$ to $F$ by examining Figure 4.4, tracing a few routes from $A$ to $F$, and somehow convincing yourself that the path you had chosen had the least cost among all possible paths. (Did you check all of the 12 possible paths between $A$ and $F$? Probably not!) Such a calculation is an example of a centralized routing algorithm—the routing algorithm was run in one location, your brain, with complete information about the network. Broadly, one way in which we can classify routing algorithms is according to whether they are global or decentralized:

Kurose at 281.

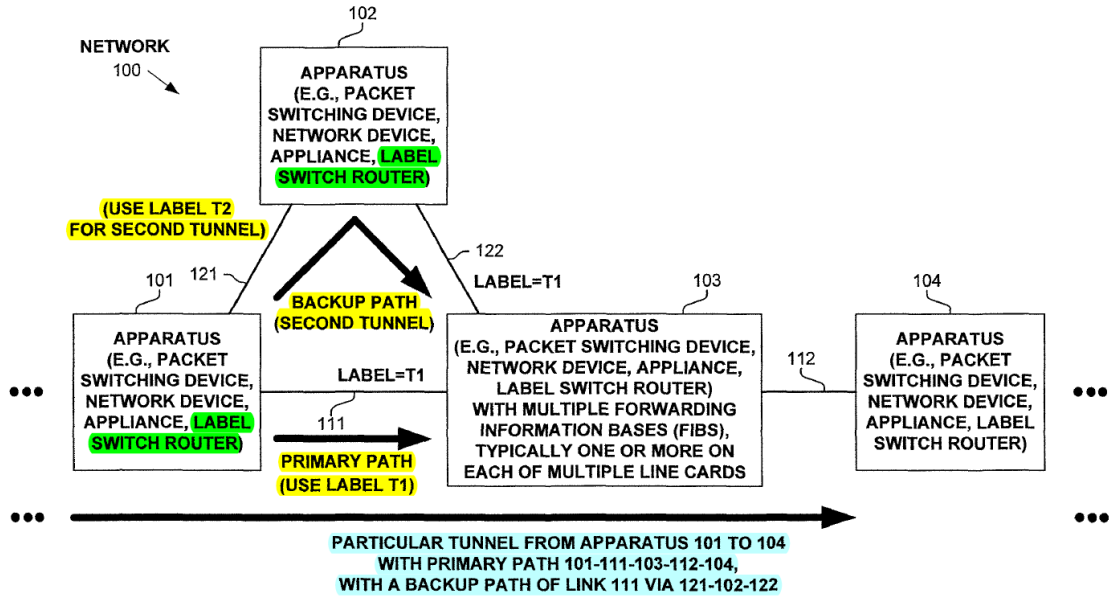
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | | "A global routing algorithm computes the least-cost path between a source and destination using complete global knowledge about the network." Kurose at 281.<br><br>"In a decentralized routing algorithm, the calculation of the least-cost path is carried out in an iterative, distributed manner." Kurose at 282. |
| 14[f] | logic configured to reselect said working entity and said protection entity from said plurality of transport entity descriptors upon a reselection event, | The Reference discloses logic configured to reselect said working entity and said protection entity from said plurality of transport entity descriptors upon a reselection event.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[d]. |
| 14[g] | wherein said reselection event is selected from a group consisting of adding an entity to said plurality of transport entities, removing an entity from said plurality of transport entities, an operational status change for one of said plurality of transport entities, and a change in overall cost for one | The Reference discloses wherein said reselection event is selected from a group consisting of adding an entity to said plurality of transport entities, removing an entity from said plurality of transport entities, an operational status change for one of said plurality of transport entities, and a change in overall cost for one of said plurality of transport entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[e]. |

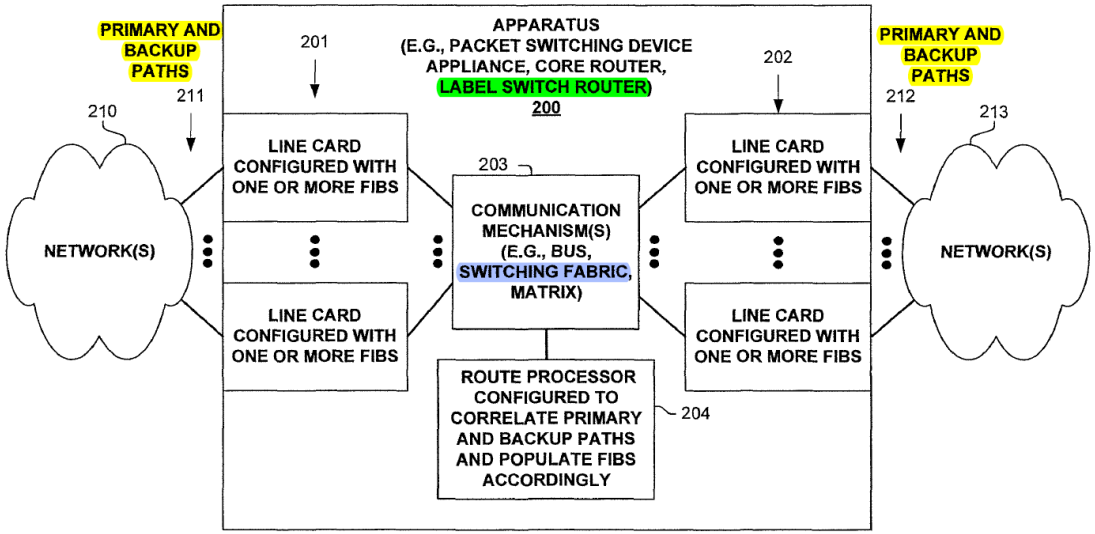
| No. | '821 Patent Claim 14 | The Reference |
|---|---|---|
| | of said plurality of transport entities. | |

| No. | '821 Patent Claim 15 | The Reference |
|---|---|---|
| 15 | The system of claim 14 wherein said entity protection switch comprises a 1:1 switch. | The Reference discloses the system of claim 14 wherein said entity protection switch comprises a 1:1 switch.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 16 | The Reference |
|---|---|---|
| 16 | The system of claim 14 wherein said entity protection switch comprises a 1+1 switch. | The Reference discloses the system of claim 14 wherein said entity protection switch comprises a 1+1 switch.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| 17[preamble] | Non-transitory computer readable media configured to perform a method comprising the steps of: | The Reference discloses non-transitory computer readable media configured to perform a method comprising the steps of.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |
| 17[a] | providing a plurality of MPLS transport entities between a first endpoint and a second endpoint; | The Reference discloses providing a plurality of MPLS transport entities between a first endpoint and a second endpoint.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[a].<br><br>Cisco created and developed the MPLS and MPLS-TE standards and patented technology based on those standards *before* Orckit utilized such technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Filsfils<br>• Taylor<br>• Vasseur '879<br>• Rustogi |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | **Filsfils discloses**:<br>"In one embodiment, forwarding information bases (FIBs) are selectively populated in a packet switch. A packet switching device determines, based on one or more protocol signaling messages, a subset, which is less than all, on which FIBs a lookup operation may be performed for identifying forwarding information for a received particular packet. The packet switching device populates each of these FIBs, but not all of the FIBs of the packet switching device, with forwarding information corresponding to the particular forwarding value. Thus, FIB resources are consumed for only those FIBs which could actually be used, and not all of the FIBs, for forwarding packets in the data plane of the packet switching device, whether these packets are received on a primary or backup path." Filsfils, Abstract.<br>  <br><br>**FIGURE 1**<br>Filsfils, FIG. 1 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIGURE 2<br>Filsfils, FIG. 2 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|

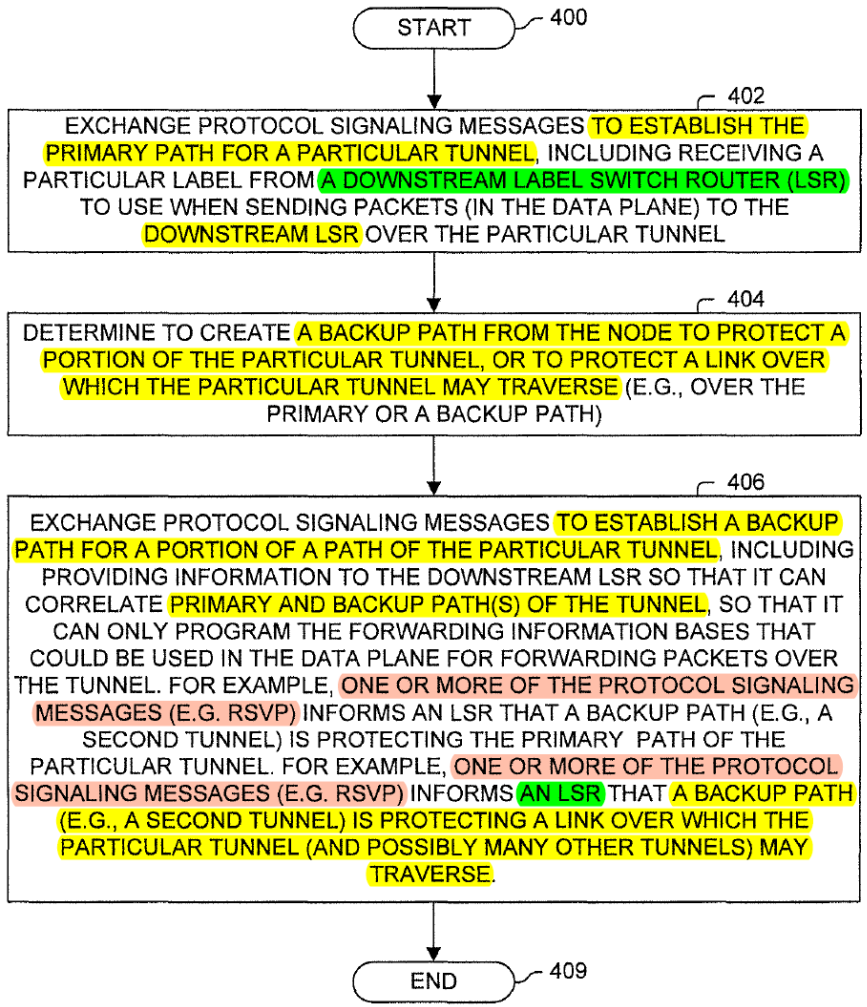
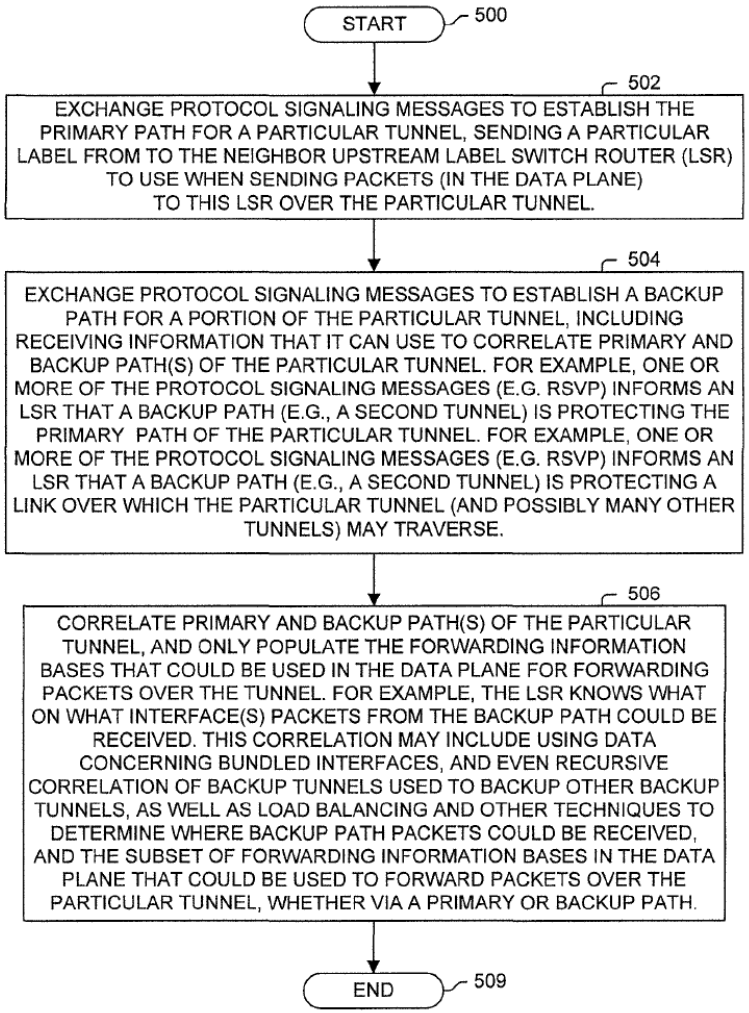



**FIGURE 4**

Filsfils, FIG. 4 (annotated).

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 5**<br>Filsfils, FIG. 5. |

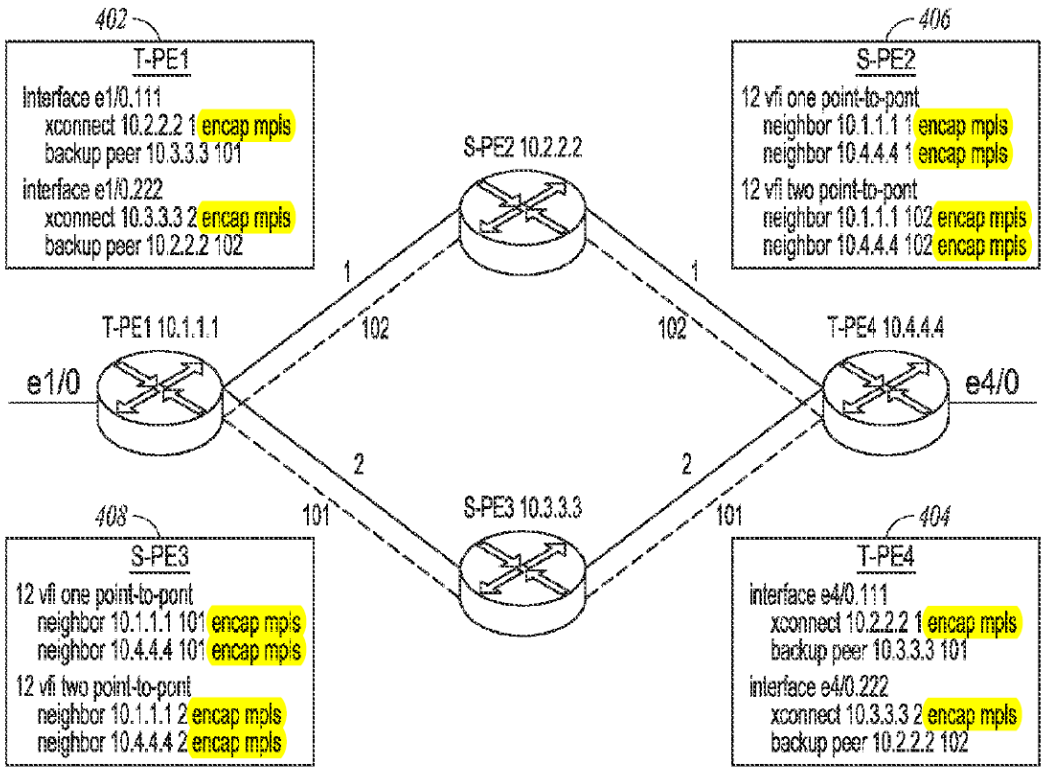
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology, including switching packets on labels especially in the core network using Multiprotocol Label Switching (MPLS)." Filsfils, 1:12-22.<br><br>"Tunnels, such as MPLS-TE (Traffic Engineering) and MPLS-TP (Transport Profile), are paths established through a network in order to transport packets efficiently through a label switched network. Fast Re-Route (FRR) is a technology that allows backup paths to be established in the network, which can be used in case of a problem with a primary path (original primary path or currently used backup path) of the tunnel. RFC 4090, entitled "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," provides an extension of the protocol signaling to establish backup label switched path (LSP) tunnels for local repair of LSP tunnels." Filsfils, 1:23-33.<br><br>"Expressly turning to the figures, FIG. 1 illustrates a network 100 operating according to one embodiment. Shown are four apparatus 101-104 (e.g., packet switching devices such as a label switch router, network device, and/or appliance). For explanation purposes, each of apparatus 101-104 will be referenced as a label switch router (LSR)." Filsfils, 5:41-46.<br><br>"As shown, a particular tunnel is established, using a signaling protocol and exchanging of protocol signaling messages. Note, LSR 101 may, or may not, be an endpoint of the particular tunnel (e.g., LSR 101 may be an intermediate LSR on the path of the particular tunnel). The primary path of the particular tunnel includes spans from LSR 101 via link 111 to LSR 103 and via link 112 to LSR 104. Note, LSR 104 may be an intermediate LSR on the path of the particular tunnel, or an endpoint of the particular tunnel. Further, for this example embodiment, LSR 103 signals LSR 101 to use label T1 at the top of the label stack in the header of a packet sent to it on the particular tunnel." Filsfils, 5:47-58.<br>"A second tunnel from LSR 101 via link 121 to LSR 102 and via link 122 to LSR 103 is similarly configured using a signaling protocol and exchanging of protocol signaling |

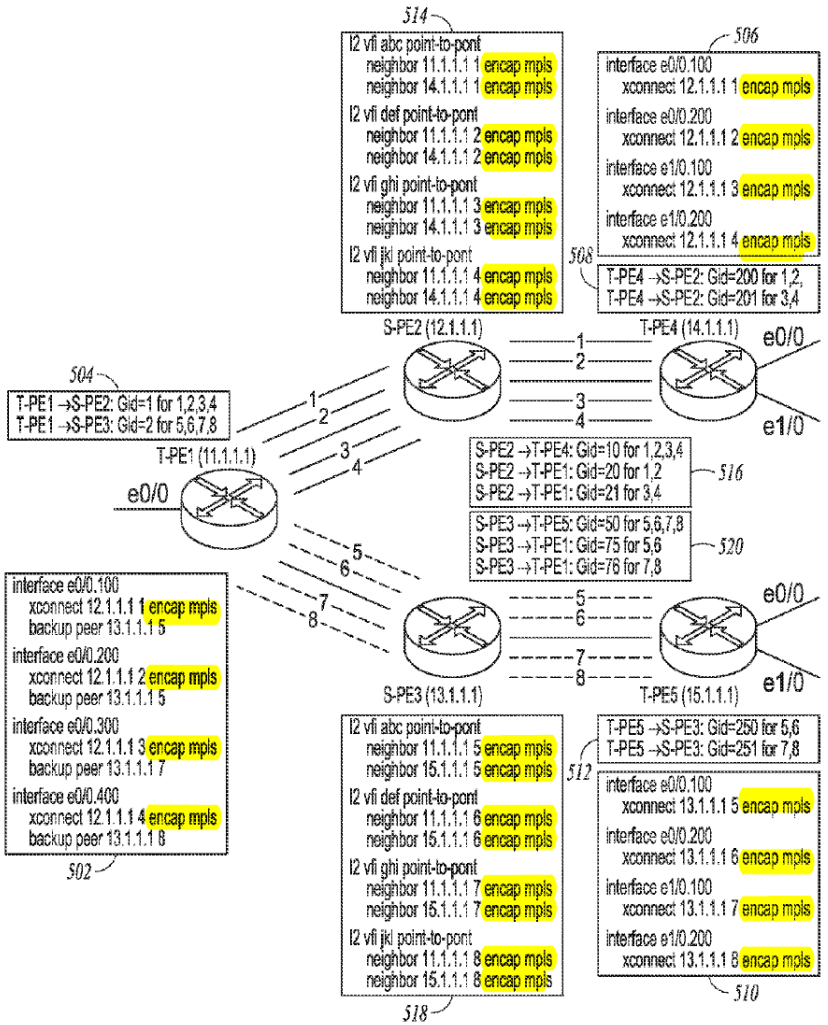
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | messages. For example purposes, LSR 102 signals LSR 101 to use label (T2) at the top of the label stack in the header of a packet sent to it on the second tunnel. In one embodiment, LSR 101 creates the second tunnel in response to determining, or being instructed to, create a backup path to protect link 111 and/or protect all or certain tunnels traversing link 111." Filsfils, 5:59-67.<br><br>"As shown in FIG. 1, link 111 (primary path of the particular tunnel and/or all or certain tunnels traversing link 111) is protected by LSR 101 using the second tunnel (backup path). When sending packets over the particular tunnel over link 111, LSR 101 includes label T1 at the top of the label stack of these packets. If link 111 cannot be used for communicating packets of the particular tunnel, LSR 101 sends packets over the backup path for the particular tunnel by sending packets to LSR 102, with these packets having a label stack including: label T2 followed by label T1. Thus, LSR 102 will receive these packets, pop the top label (T2) from the label stack of each of these packets, identify based on label T2 to send these packets to LSR 103. After popping the top label, the label at the top of the label stack of these packets is T1, which is the same label LSR 103 expects to receive for the particular tunnel. Therefore, these packets received with label T1 at the top of their label stack, are forwarded (after popping label T1 from their label stack) by LSR 103 over the particular tunnel to LSR 104." Filsfils, 6:6-24.<br><br>"One embodiment acquires such additional information by extending Resource Reservation Protocol (RSVP) to provide information which allows a packet switch to correlate primary and backup paths. Thus, a packet switch can use this additional information in determining which of its forwarding information bases (FIBs) could possibly be used in forwarding packets (e.g., in the data plane of the packet switch)." Filsfils, 6:51-57.<br><br>"In providing this additional information to LSR 103, one embodiment communicates an extended RSVP message (including a new or modified RSVP object) or other message to LSR 103 on the second tunnel. This messages designates one or more primary tunnels (e.g., label T1 in our example) and/or a link (e.g., link 111). As LSR 103 knows what interface that it received this message, LSR 103 knows that it must populate forwarding information for these primary tunnels, either specified (e.g., by a label such as T1), or all labels corresponding to |

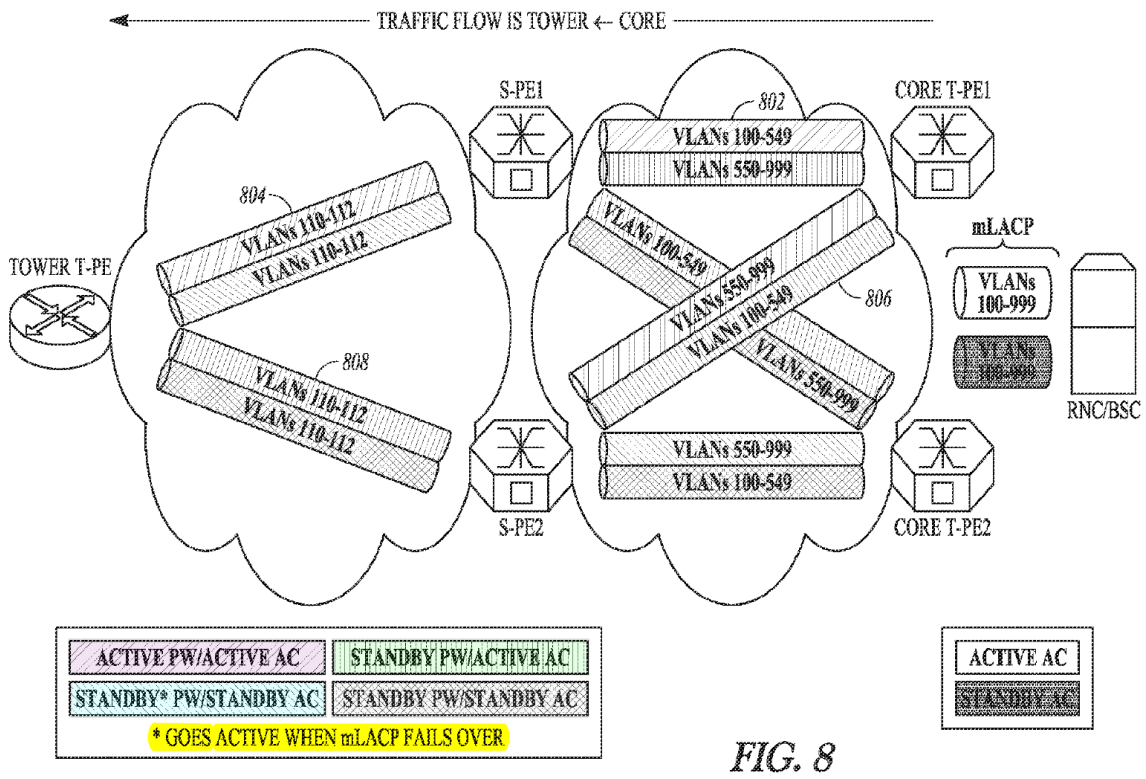
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | tunnels which could be received over link 111. In one embodiment, the extended RSVP or other message communicated to LSR 103 also includes an identification of the backup tunnel (e.g., T2) over which the RSVP or other message is being received, as the identification the tunnel over which a packet is received is often not communicated in a packet (e.g., in the case of Penultimate Hop Popping)." Filsfils, 7:19-34.<br><br>"As shown in FIG. 1, one embodiment includes apparatus 103, which populates less than all of its FIBs with forwarding information for a tunnel (although all FIBs may be populated for certain tunnels). One embodiment includes apparatus 101 and/or 102 which communicates, via a signaling protocol (e.g., an extension of RSVP, or using another protocol), information which allows apparatus 103 to determine the relationship between primary and backup paths, such that apparatus 104 can correlate this primary and backup path information (possibly also correlating backup path of backup path information, and/or bundled interfaces and/or bundled links) to identify a minimum subset of the FIBs that could possibly be used in forwarding packets of particular primary paths (e.g., tunnels)." Filsfils, 7:49-62<br><br>"Turning to FIG. 2, illustrates an apparatus 200 (e.g., packet switching devices such as a label switch router, network device, and/or appliance) operating in one embodiment. As shown, apparatus 200 includes line cards 201, 202 communicatively coupled via communication mechanism(s) 203 (e.g., bus, switching fabric, and/or matrix). Additionally, route processor 204 is configured to correlate primary and backup paths of tunnels, and to populate minimum subsets of FIBs with forwarding information for labels. Again, a minimum subset of FIBs for a particular path or label of the particular path is the set of FIBs that are determined to possibly be used in forwarding packets of a primary path, whether the label is received in a packet over the primary path or over a backup path, and possibly considering backup paths of a backup path and/or the possibly effect of bundled interfaces and/or bundled links." Filsfils, 7:63-8:11.<br><br>"As shown in FIG. 2, apparatus 200 is communicatively coupled via primary and backup paths 211, 212 to networks 210 and 213 (which could be the same network). As illustrated, each of line cards 201, 202 includes one or more FIBs. By correlating on which line card(s) 201, 202 and even within line cards 201, 202 that have multiple FIBs, primary and backup path(s) of tunnels, the number of FIB entries populated in apparatus 200 can typically be reduced, |

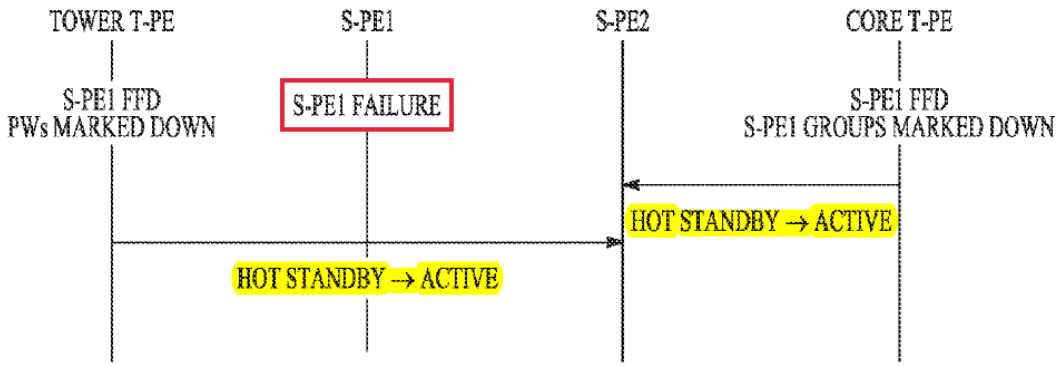
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | possibly significantly saving memory/storage resources and resources used to populate the FIBs." Filsfils, 8:12-21.<br><br>"FIG. 4 illustrates a process performed in one embodiment. Processing begins with process block 400. In process block 402, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes receiving a particular label for a downstream LSR to use when sending packets to the downstream LSR over the particular tunnel." Filsfils, 8:61-67.<br><br>"In process block 404, a determination is made to create a backup path from the node (e.g. the node performing these operations). This backup path may be used to protect one or more particular tunnels, and/or may be used to protect a link which is used to carry packet traffic of one or more tunnels." Filsfils, 9:1-5.<br><br>"In process block 406, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including providing information to the downstream LSR so that the downstream LSR can correlate primary and backup path(s) of the particular tunnel and substantially only program the FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:6-23.<br><br>"FIG. 5 illustrates a process performed in one embodiment. Processing begins with process block 500. In process block 502, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes sending a particular label for an upstream LSR to use when sending packets over the particular tunnel to this apparatus (e.g., an LSR performing these operations)." Filsfils, 9:26-32. |

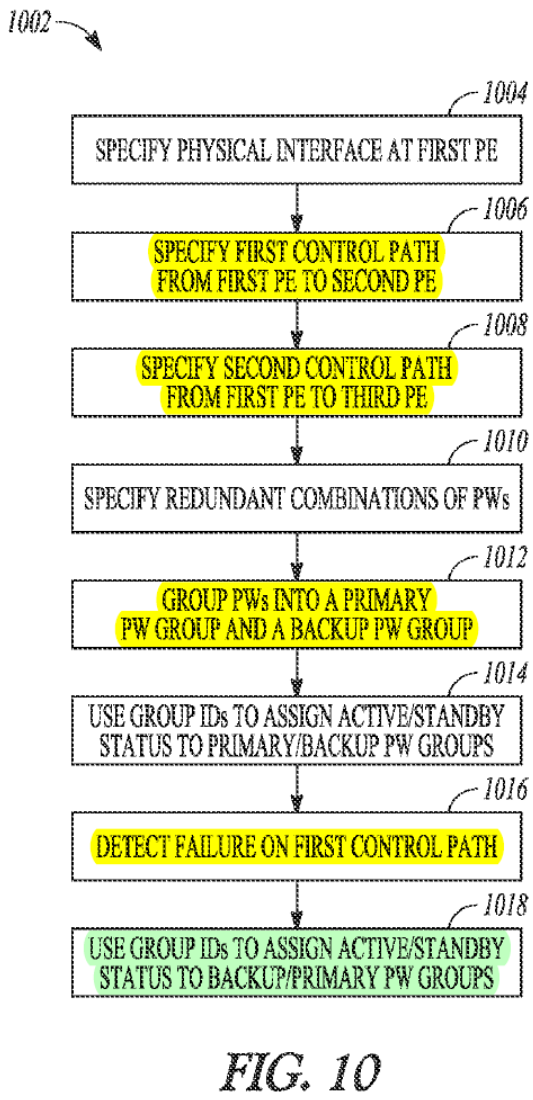
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "In process block 504, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including receiving information that the LSR can use to correlate primary and backup path(s) of the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:33-47.<br><br>"In process block 506, the primary and backup path(s) of the particular tunnel are correlated to identify the set of FIBs that could possibly be used in forwarding packets of the particular tunnel. Substantially only those FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel (either through a primary or backup path) are populated with the forwarding information (e.g., an entry corresponding to the label it advertised to use for the particular tunnel) for the particular tunnel. For example, the LSR knows what on what interface(s) packets from the backup path could be received. This correlation may include using data concerning bundled interfaces, and even recursive correlation of backup tunnels used to backup other backup tunnels, as well as load balancing and other techniques to determine where backup path packets could be received, and the subset of forwarding information bases in the data plane that could be used to forward packets over the tunnel, whether via a primary or backup path." Filsfils, 9:48-65.<br><br>**Taylor discloses**:<br>"Grouping pseudowires based on hardware interfaces and configured control paths enables improved pseudowire failover performance. Signaling status changes (e.g., from standby to active status) is facilitated by using group IDs for the pseudowire groups, thereby enabling improved failover performance when there is disruption in the network." Taylor, Abstract. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Taylor, FIG. 4 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 5*<br>Taylor, FIG. 5 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br>Taylor, FIG. 8 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Taylor, FIG. 9 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Taylor, FIG. 10 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "The present disclosure relates generally to communication networks and more particularly to pseudowire configurations in communication networks." Taylor, 1:8-10.<br><br>"Virtual Private Network (VPN) services provide secure network connections between different locations. A company, for example, can use a VPN to provide secure connections between geographically dispersed sites that need to access the corporate network so that each customer edge (CE) end point or node can communicate directly and independently with all other CE nodes. Different types of VPNs have been classified by the network layer used to establish the connection between the customer and provider network. For example, Virtual Private LAN Service (VPLS) is an architecture that delivers a multipoint Layer 2 VPN (L2VPN) service that in all respects emulates an Ethernet Local Area Network (LAN) across a wide metropolitan geographic area. All services in a VPLS appear to be on the same LAN, regardless of location. In other words, with VPLS, customers can communicate as if they were connected via a private Ethernet segment, i.e., multipoint Ethernet LAN services." Taylor, 1:12-28.<br><br>"In this context, each CE device at a customer site is connected to the service provider network at a provider edge PE) device by an Attachment Circuit (AC) that provides the customer connection to a service provider network, that is, the connection between a CE node and its associated PE node. Within the provider network, each PE device includes a Virtual Switch Instance (VSI) that emulates an Ethernet bridge (i.e., switch) function in terms of Media Access Control (MAC) address learning and forwarding in order to facilitate the provisioning of a multipoint L2VPN. A pseudowire (PW) is a virtual connection between two PE devices that connect two attachment circuits. In the context of the VPLS service, a pseudowire can be thought of as a point-to-point virtual link for each offered service between a pair of VSIs. Therefore, if each VSI can be thought of as a virtual Ethernet switch for a given customer service instance, then each pseudowire can be thought of as a virtual link connecting these virtual switches to each other over a Packet Switched Network (PSN) for that service instance." Taylor, 1:29-47. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "Since the failure of pseudowires obviously degrades network performance, some effort has been directed towards adding system redundancies including redundant pseudowires. However, the presence of redundant pseudowires alone is insufficient to improve overall failover performance, that is, the ability to switch over automatically to a redundant or backup system. Thus, there is a need for improved methods for managing pseudowires to facilitate pseudowire switching and enable improved failover performance." Taylor, 1:48-56.<br><br>"FIG. 4 shows details for PW connectivity in an exemplary network for an example embodiment." Taylor, 1:66-67.<br><br>"FIG. 5 shows details for PW grouping for an example embodiment." Taylor, 2:1-2.<br><br>"FIG. 8 shows an example network including redundant PW connections for an example embodiment." Taylor, 2:9-10.<br><br>"FIG. 9 shows an example sequence diagram for a failure mode related to the embodiment shown in FIG. 8." Taylor, 2:11-12.<br><br>"FIG. 10 shows a flowchart that illustrates a method of providing improved PW grouping according to an example embodiment." Taylor, 2:13-15.<br><br>"According to one embodiment, a method of providing improved pseudowire performance includes specifying a physical interface at a first PE node in a network, a first control path from the first PE node to a second PE node in the network, and a second control path from the first PE node to a third PE node in the network. With these specifications, the method then includes specifying redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node, and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Then these pseudowires can be grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. Group identifiers for the primary PW group and the backup PW group can then be used to assign an |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths. The method may then include detecting a failure on the first control path, and in response to the detected failure, using the group identifiers to assign the active status to the backup pseudowires and the standby status to the primary pseudowires." Taylor, 2:27-52.<br><br>"Pseudowires are used in pseudowire emulation edge-to-edge to provide a Layer 2 Virtual Private Network (L2VPN) connection. When large numbers (e.g., 4,000-6,000) of pseudowires are aggregated together on a single router, failure performance tends to be linear or O(n) where n is the number of pseudowires. While O(n) performance may be acceptable for small numbers of pseudowires, the effect on network outages can be increasingly undesirable as the number of pseudowires increases." Taylor, 2:54-62.<br><br>"For example, a cell-site router will typically start an approximately 2-minute procedure if contact with its controller, which is reached via a pseudowire, is lost for more than some threshold amount (e.g., between approximately 0.75 and 1.75 seconds in some cases). This can be a major impediment to the scalability of pseudowire deployments. These issues have become increasingly relevant as providers of Multiservice Broadband Networks (MBNs) are rapidly replacing or augmenting their traditional Synchronous Optical Networking (SONET) equipment with cheaper Ethernet equipment in the evolution towards a 4G (i.e., 4$^{th}$ generation) network." Taylor, 2:63-3:7.<br><br>"One aspect of a solution to the problem of pseudowire failure is the deployment of redundant pseudowires. For example, redundant pseudowires have been used in the context of Multiprotocol Label Switching (MPLS) networks, which use a Label Distribution Protocol (LDP) to manage labels for forwarding traffic between routers. In this context, general requirements for redundancy schemes have been developed so that duplicate pseudowires are available when a given pseudowire fails (e.g., by using active/standby status indicators). In addition, more specific implementations for redundant pseudowires have also been developed." Taylor, 3:8-18. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 1 shows a reference network model 102 with applications to example embodiments disclosed herein. The reference network model 102 includes an aggregation network 104 of PE nodes and a distribution network 106 of PE nodes between a radio network controller (RNC) (or base station controller (BSC)) 108 on the core side of the model 102 and a radio tower 111 on the tower side of the model 102. Switching provider edge nodes S-PE1 and S-PE2 connect the two networks 104, 106. On the core side, two core terminating provider edges T-PE1 and T-PE2 connect to the RNC/BSC 108 through attachment circuits 110, 112. On the tower side, one tower terminating provider edge T-PE connects to the radio tower 111 through an attachment circuit 114." Taylor, 3:19-31.<br><br>"Additionally as noted in FIG. 1, peer-PE monitoring is carried out within each network 104, 106. That is, there is peer-PE monitoring between provider edges that share a segment, for example, by multi-hop bidirectional forwarding detection (BFD). Alternatively, peer monitoring can be accomplished by other means (e.g., MPLS-TP (Transport Protocol) keep-olives). This peer-PE monitoring is used to provide the mechanism for fast failure detection. Once a failure is detected, the network can react by "rerouting" the failed pseudowires to pre-provisioned backup paths and thus provide a minimal disruption in service to the end-user. This rerouting can be accomplished by LDP signaling between provider edges." Taylor, 3:32-44.<br><br>"The reference network model 102 may be considered as part of a larger hub-and-spoke model as shown in FIG. 2. A hub-and-spoke distribution model 202 includes a core network 204, distribution networks 206, and aggregation networks 208. Network elements including distribution nodes, aggregation nodes, and towers are also shown with nominal count values (e.g., 30 distribution nodes between the core network 204 and a distribution network 206). In this model 202, tower T-PEs are the spokes white core-PEs constitute the hub. Dozens to hundreds of tower T-PEs connect to a few S-PEs; these S-PEs are quite similar to ASBRs as they act as forwarders between the two distinct MPLS domains, providing isolation and, in the case of mobility, aggregation services. Typically, several aggregation networks 208 are connected to a single distribution network 206, eventually connecting the tower with the core router that connects the tower's ACs to the RNC/BSC. There are typically several distribution |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | networks in a Radio Access Network (RAN) connected to the service provider's core Internet Protocol (IP) network." Taylor, 4:11-30.<br><br>"With reference to FIG. 1, FIG. 3 shows a variety of failure modes encountered in the reference network model 102. Failure 302 of communications between tower T-PE and the S-PE can be detected via peer monitoring when both the S-PE and the T-PE are still active/alive. For example, this failure may be due to a toss of connectivity when the BFD session goes down. Failure 304 of S-PE1 can be due to a hardware failure, power outage, or the lack of BED-session maintenance capability (e.g., if the BFD-session hello timers cannot be serviced for the prescribed period of time). Failure 306 of communications between S-PE1 and core T-PE1 can be detected via peer monitoring when both S-PE1 and core T-PE1 are still active/alive. Failure 308 at core T-PE1 can be due to a hardware failure, power outage, or the lack of BFD-session maintenance capability." Taylor, 4:46-60.<br><br>"Pseudowire connectivity is further illustrated in FIG. 4 where the illustrated network includes four nodes: T-PE1 (10.1.1.1), S-PE2 (10.2.2.2), S-PE3 (10.3.3.3), and T-PEA (10.4.4.4). For the terminating nodes T-PE1 and T-PE4, specifications for VLANs (virtual Local Area Networks) connections (i.e., pseudowires) are shown using the Internet Operating System Command Line Interface (IOS CLI). The specification 402 for T-PE1 defines two VLANS as primary/backup combinations of virtual circuits for the network. The first three lines of the specification 402 define "VLAN 111" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 111" in the first line. The second line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE2 (10.12.2) with a virtual circuit Identification (VCID) set as VCID=1, and the third line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=101. The next three lines of the specification 402 define "VLAN 222" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 222" in the fourth line. The fifth line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=2, and the sixth line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE2 (10.2.2.2) with VCID=102. These virtual circuits, VCID=1, VCID=2, VCID=101 and VCID=102 are shown in the figure between T-PE1 and the S-PEs with a solid line for the primary circuits VCID=1 |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | and VCID=2 and a dashed line for the backup circuits VCID=101 and VCID=102." Taylor, 4:61-5:23.<br><br>"With respect to T-PE1 in FIG. 4, although "VLAN 111" and "VLAN 222" share the same hardware port, they do not share the same "control path disposition." That is, "VLAN 111" is primary to S-PE2 (VCID=1) and standby to S-PE3 (VCID=101), white "VLAN 222" has an opposite configuration since it is primary to S-PE3 (VCID=2) and standby to S-PE2 (VCID=102). As discussed below, certain embodiments group pseudowires according to "control path disposition" (e.g., xconnect configuration as well as the hardware interface in order to improve failover performance. That is, to deal with both hardware port failures and switching path failures, the grouping criteria also considers the cross connects. In this case, on T-PE1 as well as T-PE4, there would exist two groups: one for active to S-PE2 and standby to S-PE3 and another for active to S-PE3 and standby to S-PE2 (i.e., the inverse configuration)." Taylor, 5:61-6:9.<br><br>"First, local connectivity is characterized by local group identifications (Group-IDs), which depend on whether the allocation is done at a T-PE or S-PE. FIG. 5 shows an embodiment that illustrates an allocation of local group IDs in a network including terminating nodes T-PE1 (11.1.1.1), T-PE4 (14.1.1.1), and T-PE5 (15.1.1.1) and switching nodes S-PE2 (12.1.1.1) and S-PE3 (13.1.1.1). The specification 502 for T-PE1 determines corresponding local group IDs 504 based on the hardware interface and the control path. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, and the Group-ID=2 corresponds to VCID=5, VCID=6, VCID=7, and VCID=8." Taylor, 6:10-21.<br><br>"Local group IDs are maintained in a database so that pseudowire redundancy is also maintained. First, in a case without pseudowire redundancy, all the xconnect configurations from the same physical interface to the same peer are assigned the same local group ID. So, for example, in Ethernet cases all xconnect configurations under sub-interfaces of the same physical interface to the same peer will be assigned the same local group ID (e.g., e0/0 and e0/1 are sub-interfaces of e0). FIG. 6 shows a database representation for T-PE4 from FIG. 5. From the root node 602 for T-PE4, there is a first interface node 604 for e0/1 and a second interface node 606 for e1/0. The first interface node 604 is configured towards a single peer |

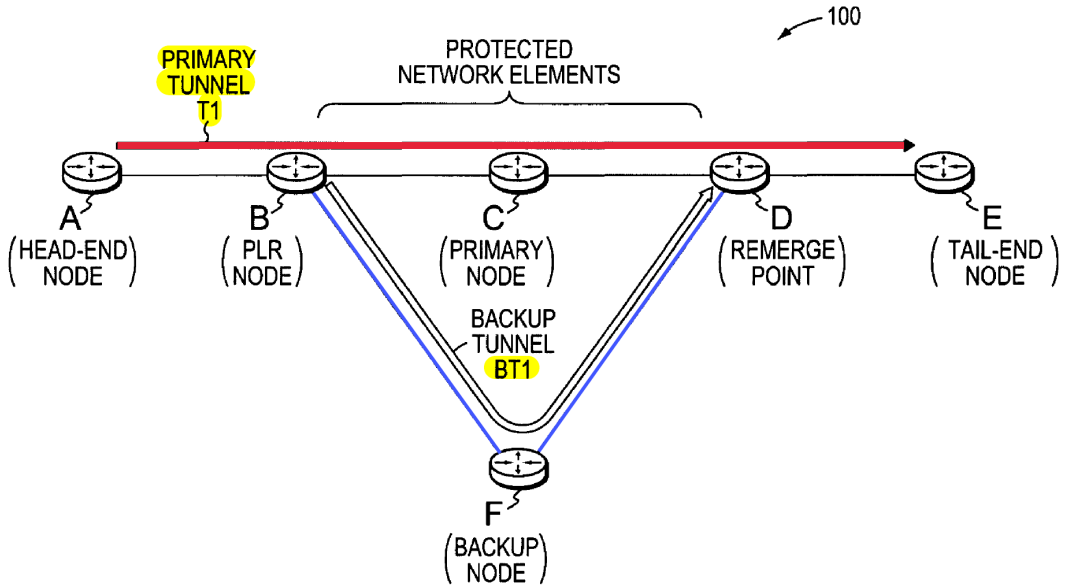
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | node (12.1.1.1) 608 and is thus assigned a single local group ID (Group-ID=200) 610. Similarly, the second interface node 606 is configured towards a single peer node (12.1.1.1) 612 and is thus assigned a single local group ID (Group-ID=201) 614. In this case from the assignment of local group IDs 508 in FIG. 5, Group-ID=200 corresponds to VCID=1 and VCID=2, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). Both pseudowires (VCID=1 and VCID=2) are assigned the same local group ID (Group-ID=200) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly, from the assignment of local group IDs 508 in FIG. 5, T-PE4 has VCID=3 and VCID=4 under the physical interface e1/0 going to the same peer S-PE2 (12.1.1.1), and the local Group ID (Group-ID=201) is assigned to these VCs." Taylor, 6:48-7:8.<br><br>"For the pseudowire redundancy case, a separate redundancy-group database is maintained by the xconnect application. This redundancy-group database contains the peer IDs in the group and the local group IDs advertised to them. This is needed to maintain a 1:1 mapping between the primary pseudowires and their corresponding backup pseudowires. FIG. 7 shows a database representation for T-PE1 from FIG. 5. From the root node 702 for T-PE1, there is an interface node 704 for e0/0 and a redundancy group node 706 that shows connections for configurations to a first peer node (12.1.1.1) 708, which is assigned a local group ID (Group-ID=1) 710, and a second peer node (13.1.1.1) 712, which is assigned a local group ID (Group-ID=2) 714. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). These pseudowires are assigned the same group ID (Group-ID=1) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly from the assignment of local group IDs 504 in FIG. 5, T-PE1 has VCID=5, VCID=6, VCID=7, and VCID=8 under the physical interface e0/0 going to another peer S-PE3 (13.1.1.1), and the local group ID (Group-ID=2) is assigned to these VCs. In this case these local group IDs are organized as a redundancy group 706." Taylor, 7:9-35. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "Pseudowire grouping allows multiple pseudowires to be aggregated together when signaling either label withdrawals or status changes between segment end-point provider edges. This signaling can be carried out through LDP grouping TLV (Type Length Value). For example, when a PE node fails (e.g., failure 304 of S-PE1), aggregating the inter-segment PE signaling using the grouping TLV can provide significant scaling advantages. This allows all pseudowires sharing a physical port and PW configuration (e.g., xconnect configuration) to be signaled en masse between segment-adjacent provider edges." Taylor, 7:36-46.<br><br>"FIG. 8 shows an example based on FIG. 1 where VLAN ACs are shown as grouped by both port/HW-interface and pseudowire-class. The grouping criterion allows all "similar" pseudowires to be signaled together: All the grouped pseudowires share the same port and next-hop provider edge. Additionally, the figure contains many pseudowires, each grouped into a shaded tube. For example, the tube labeled "VLANs 100-549" contains 450 pseudowires grouped together. This figure depicts an incoming Ethernet comprised of 900 VLANs being segmented in two with 450 VLANs (100-549) active to S-PE1 while the other half of the VLANs (550-999) being active to S-PE2. This might be considered a type of manual load balancing. Furthermore, the aggregation network is only showing a single tower and the VLANs associated with it; other VLAN destinations are not shown in the figure." Taylor, 7:47-62.<br><br>"'VLANs 110-112' are active along a first pseudowire path 802 from Core T-PE1 to S-PE1 and a second pseudowire path 804 from S-PE1 to Tower T-PE. When a failure occurs at S-PE1 (e.g., as the switching node failure 304 shown in FIG. 3), then the standby pseudowires become active for 'VLANs 110-112' along a first pseudowire path 806 from Core T-PE1 to S-PE2 and a second pseudowire path 808 from S-PE2 to Tower T-PE." Taylor, 7:63-8:3.<br><br>"The standby pseudowires in FIG. 8 can be configured as HSPWs, a configuration that enables ACs to quickly failover to pre-provisioned pseudowires that are in active state but set to not-forwarding. Then when a failure occurs, switching over to these pre-provisioned HSPWs occurs quickly by switching from not-forwarding status to forwarding status." Taylor, 8:4-9. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 9 shows a UML (Unified Modeling Language) sequence diagram of signaling events related to the failover procedure illustrated in FIG. 8 for a switching node failure 304. The Tower T-PE and the Core T-PE independently detect a failure at S-PE1 (e.g., BFD forwarding with LDP signaling), and then mark down the status of the currently active pseudowires routed through S-PE1 and mark up the status of the standby pseudowires routed through S-PW2. Other failure modes shown in FIG. 3 can be handled similarly." Taylor, 8:10-18.<br><br>"In general, it is desirable for MPLS-based. Ethernet networks to react quickly to failures, so proactive detection mechanisms are employed in order to pick up system failures quickly. All proactive monitoring is typically done between PE peers on a single MPLS network. These provider edges on the edges of the MPLS network act similarly to an Autonomous System Boundary Router (ASBR). As a result, related embodiments detect control path failures, which may not be the same as pseudowire data path failures. That is, the data packets and control packets may take different paths between provider edges in a MPLS network although typically these paths are coincident. Thus, when the control and data paths are not coincident, if the control path fails, then all pseudowires utilizing the control path are marked as failed. As a corollary, if the data path fails and the control path remains healthy, then failure will not be detected from monitoring the control path." Taylor, 8:19-35.<br><br>"A failure of a monitored provider edge initiates a switchover of all active pseudowires using the failing provider edge to their configured HSPWs (if they exist). Grouping can greatly reduce the number of messages needed between provider edges (Inter-PE Aggregation) and within a single provider edge (Intra-PE Aggregation). Furthermore, the MPLS network itself may be internally resilient deploying technologies such as, but not limited to, MPLS-TE (MPLS Traffic Engineering) and ERR (Fast Reroute). The paths across the MPLS network may recover quickly and might not trip the fault-monitoring systems." Taylor, 8:36-46.<br><br>"With reference to the above discussion, FIG. 10 shows a method 1002 of providing improved PW grouping according to an example embodiment. In a first operation 1004 of the method 1002, a physical interface is specified at a first PE node in a network. In a second operation 1006, a first control path is specified from the first PE node to a second PE node in the network. In a third operation 1008, a second control path is specified from the first PE node to a third |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | PE node in the network. These control paths related to a common physical interface can be used to characterize redundant pairs of pseudowires." Taylor, 8:48-58.<br><br>"In a fourth operation 1010, redundant combinations of pseudowires are specified, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Although a redundant combination may relate a single backup pseudowire to a given primary pseudowire, in some cases multiple backup pseudowires will be related to a given primary pseudowire for increased redundancy. In a fifth operation 1012, these pseudowires are grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. In a sixth operation 1014, group identifiers for the primary PW group and the backup PW group are used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 8:59-9:11.<br><br>"In an optional seventh operation 1016, a failure may be detected on the first control path, and in an optional eighth operation 1018, in response to the detected failure, the group identifiers may be used to assign the active status to the backup pseudowires and the standby status to the primary pseudowires. For example, the failure on the first control path may be detected by using BED packet streams between PE nodes of the network. Then the detected failure can be signaled to PE nodes in the network by sending LDP status updates between PE nodes in the network. Then, after receiving the failure detection signals, the group identifiers can be used again to assign the active status to the backup pseudowires and the standby status to the primary pseudowires by sending LDP status updates between PE nodes in the network." Taylor, 9:12-26.<br><br>"Typically the network in is an MPLS network and the PE nodes are routers that provide network services to connected CE nodes of a customer network. In general, each control path is an Internet Protocol (IP) routing path between PE nodes in the network and each data path is a label switched path (LSP) between PE nodes in the network." Taylor, 9:27-32. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 11 shows a schematic representation of an apparatus 1102, in accordance with an example embodiment. For example, the apparatus 1102 may be used to implement the method 1002 of providing improved pseudowire grouping as described above with reference to FIG. 10. The apparatus 1102 is shown to include a processing system 1104 that may be implemented on a server, client, or other processing device that includes an operating system 1106 for executing software instructions." Taylor, 10:2-10.<br><br>"In accordance with an example embodiment, the apparatus 1102 includes a PW management module 1108 that includes a first specification module 1110, a second specification module 1112, third specification module 1114, a fourth specification module 1116, a grouping module 1118, and an assignment module 1120. The first specification module 1110 operates to specify a physical interface at a first PE node in a network. The second specification module 1112 operates to specify a first control path from the first PE node to a second PE node in the network. The third specification module 1114 operates to specify a second control path from the first PE node to a third PE node in the network. The fourth specification module 1116 operates to specify redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node." Taylor, 10:11-29.<br><br>"The grouping module 1118 operates to group the pseudowires into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. The assignment module 1120 operates to use group identifiers for the PW groups to assign an active status to the primary pseudowires and a standby status to the backup pseudowires. The active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 10:30-38. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | **Vasseur '879 discloses**:<br>"A technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, Abstract.<br>  <br><br>FIG. 1<br>Vasseur '879, FIG. 1 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|

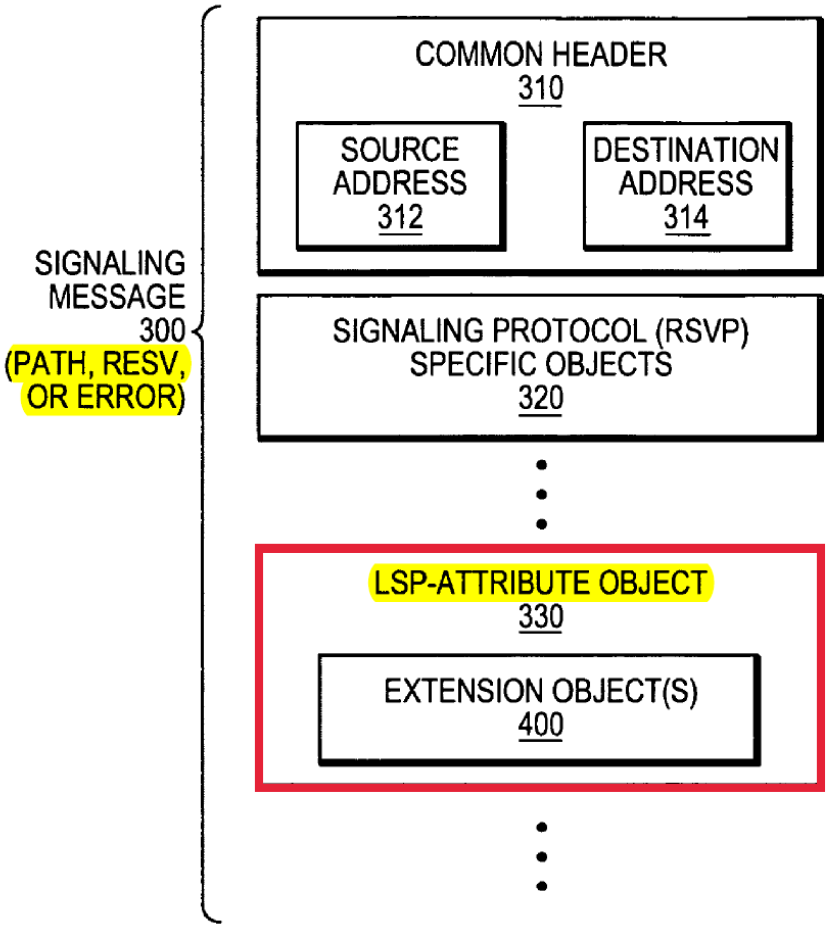
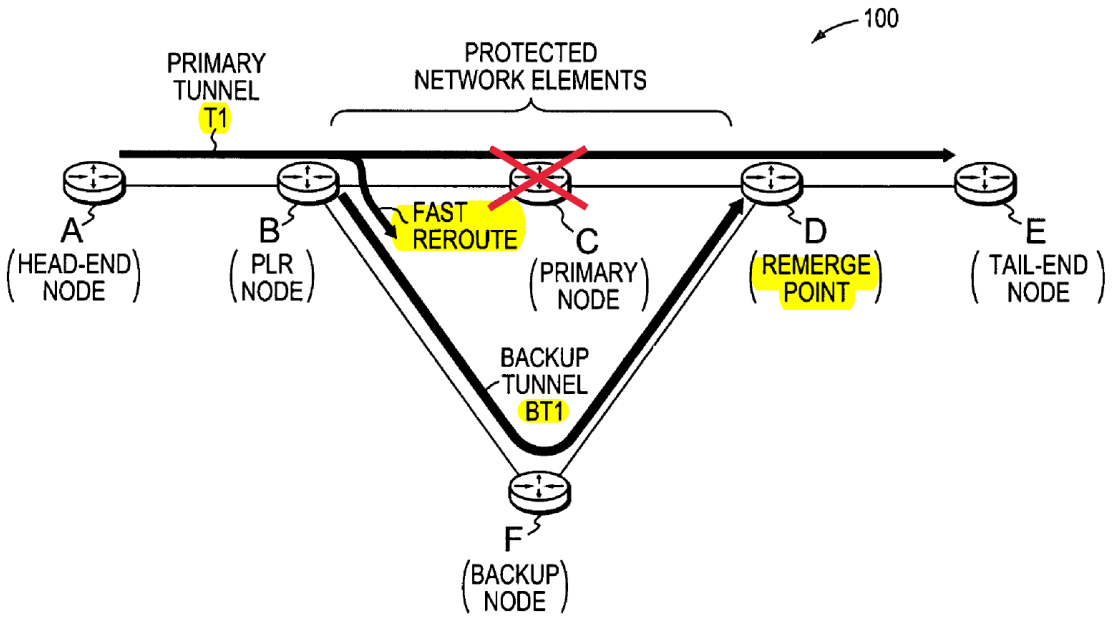
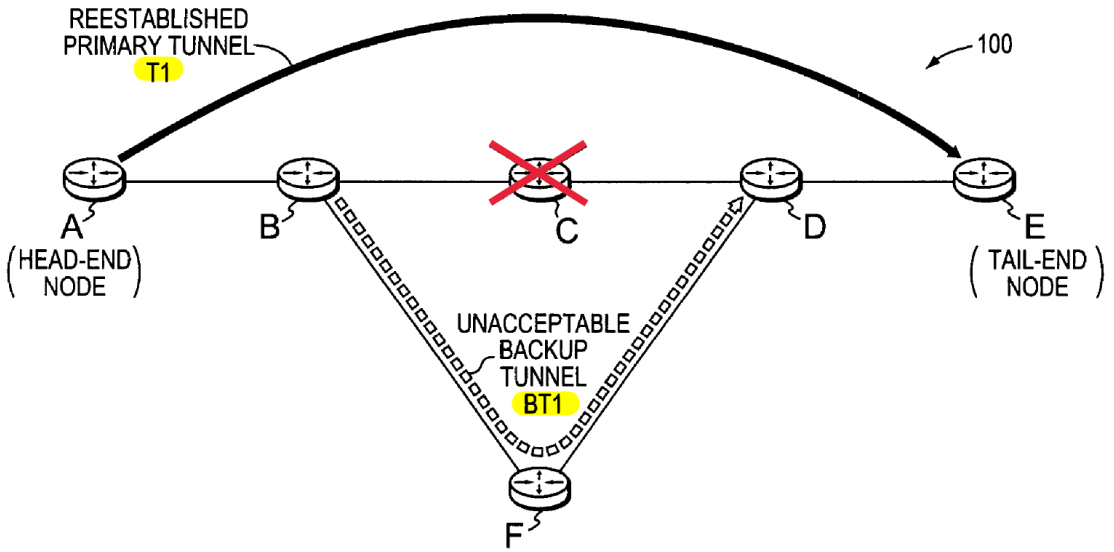
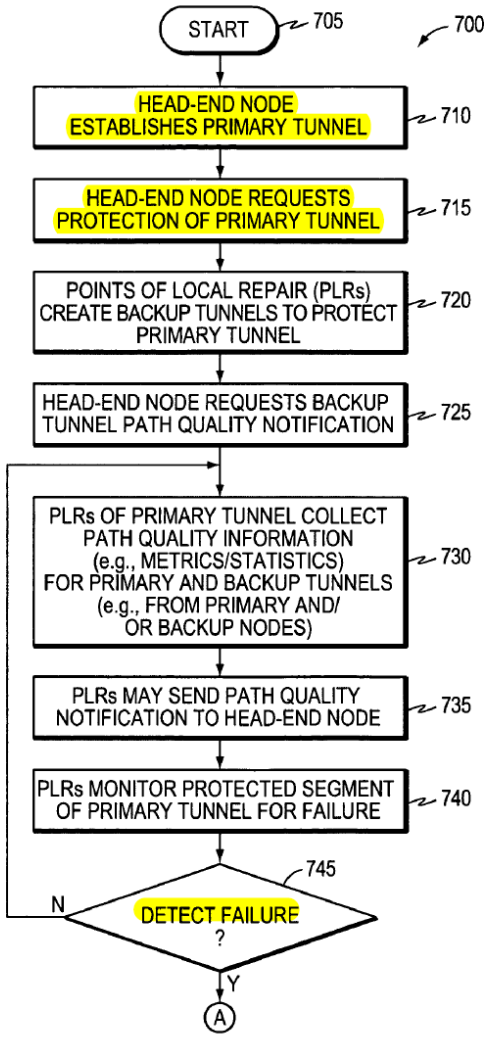


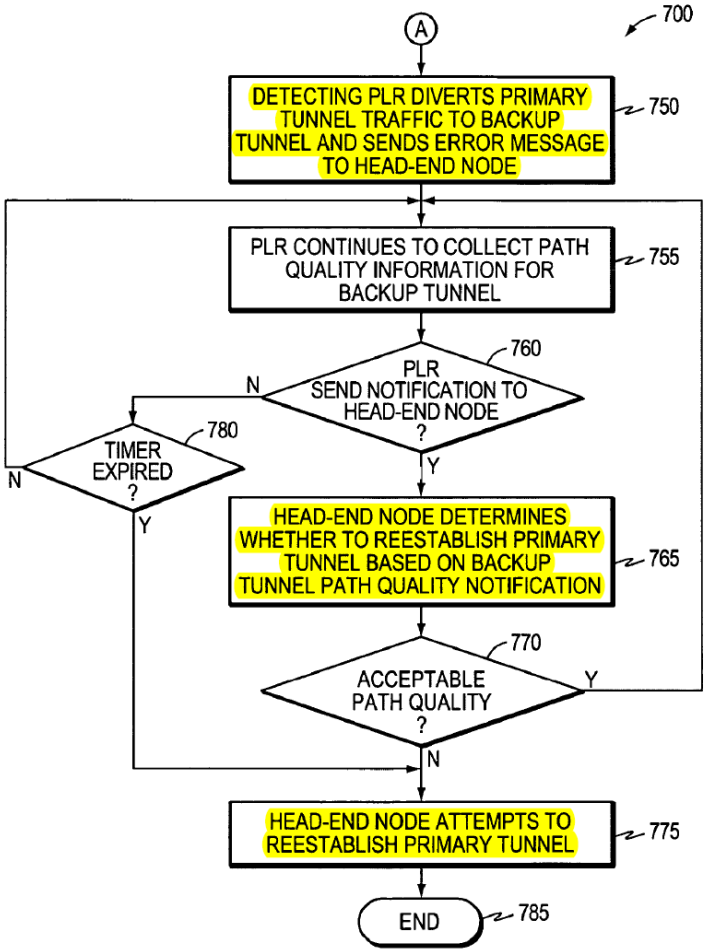

FIG. 3

Vasseur '879, FIG. 3 (annotated).

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Vasseur '879, FIG. 5 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br>Vasseur '879, FIG. 6 (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 7A<br>Vasseur '879, FIG. 7A (annotated). |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 7B<br><br>Vasseur '879, FIG. 7B. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "The present invention relates to computer networks and more particularly to dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 1:10-14.<br><br>"Since management of interconnected computer networks can prove burdensome, smaller groups of computer networks may be maintained as routing domains or autonomous systems. The networks within an autonomous system (AS) are typically coupled together by conventional "intradomain" routers configured to execute intradomain routing protocols, and are generally subject to a common authority. To improve routing scalability, a service provider (e.g., an ISP) may divide an AS into multiple "areas." It may be desirable, however, to increase the number of nodes capable of exchanging data; in this case, interdomain routers executing interdomain routing protocols are used to interconnect nodes of the various ASes. Moreover, it may be desirable to interconnect various ASes that operate under different administrative domains. As used herein, an AS or an area is generally referred to as a "domain," and a router that interconnects different domains together is generally referred to as a 'border router.'" Vasseur '879, 1:40-56.<br><br>"An example of an interdomain routing protocol is the Border Gateway Protocol version 4 (BGP), which performs routing between domains (ASes) by exchanging routing and reachability information among neighboring interdomain routers of the systems. An adjacency is a relationship formed between selected neighboring (peer) routers for the purpose of exchanging routing information messages and abstracting the network topology. The routing information exchanged by BGP peer routers typically includes destination address prefixes, i.e., the portions of destination addresses used by the routing protocol to render routing ("next hop") decisions. Examples of such destination addresses include IP version 4 (IPv4) and version 6 (IPv6) addresses. BGP generally operates over a reliable transport protocol, such as TCP, to establish a TCP connection/session. The BGP protocol is well known and generally described in Request for Comments (RFC) 1771, entitled A Border Gateway Protocol 4 (BGP-4), published March 1995." Vasseur '879, 1:57-2:7. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "Examples of an intradomain routing protocol, or an interior gateway protocol (IGP), are the Open Shortest Path First (OSPF) routing protocol and the Intermediate-System-to-Intermediate-System (IS-IS) routing protocol. The OSPF and IS-IS protocols are based on link-state technology and, therefore, are commonly referred to as link-state routing protocols. Link-state protocols define the manner with which routing information and network-topology information are exchanged and processed in a domain. This information is generally directed to an intradomain router's local state (e.g., the router's usable interfaces and reachable neighbors or adjacencies). The OSPF protocol is described in RFC 2328, entitled OSPF Version 2, dated April 1998 and the IS-IS protocol used in the context of IP is described in RFC 1195, entitled Use of OSI IS-IS for routing in TCP/IP and Dual Environments, dated December 1990, both of which are hereby incorporated by reference." Vasseur '879, 2:8-24.<br><br>"An intermediate network node often stores its routing information in a routing table maintained and managed by a routing information base (RIB). The routing table is a searchable data structure in which network addresses are mapped to their associated routing information. However, those skilled in the art will understand that the routing table need not be organized as a table, and alternatively may be another type of searchable data structure. Although the intermediate network node's routing table may be configured with a predetermined set of routing information, the node also may dynamically acquire ("learn") network routing information as it sends and receives data packets. When a packet is received at the intermediate network node, the packet's destination address (e.g., stored in a header of the packet) may be used to identify a routing table entry containing routing information associated with the received packet. Among other things, the packet's routing information indicates the packet's next-hop address." Vasseur '879, 2:25-41.<br><br>"Multi-Protocol Label Switching (MPLS) Traffic Engineering has been developed to meet data networking requirements such as guaranteed available bandwidth or fast restoration. MPLS Traffic Engineering exploits modem label switching techniques to build guaranteed bandwidth end-to-end tunnels through an IP/MPLS network of label switched routers (LSRs). These tunnels are a type of label switched path (LSP) and thus are generally referred to as MPLS Traffic Engineering (TE) LSPs. Examples of MPLS TE can be found in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels dated December 2001, RFC 3784 |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | entitled Intermediate-System-to-Intermediate-System (IS-IS) Extensions for Traffic Engineering (TE) dated June 2004, and RFC 3630, entitled Traffic Engineering (TE) Extensions to OSPF Version 2 dated September 2003, the contents of all of which are hereby incorporated by reference in their entirety." Vasseur '879, 2:58-3:6.<br><br>"Establishment of an MPLS TE-LSP from a head-end LSR to a tail-end LSR involves computation of a path through a network of LSRs. Optimally, the computed path is the "shortest" path, as measured in some metric, that satisfies all relevant LSP Traffic Engineering constraints such as e.g., required bandwidth, "affinities" (administrative constraints to avoid or include certain links), etc. Path computation can either be performed by the head-end LSR or by some other entity operating as a path computation element (PCE) not co-located on the head-end LSR. The head-end LSR (or a PCE) exploits its knowledge of network topology and resources available on each link to perform the path computation according to the LSP Traffic Engineering constraints. Various path computation methodologies are available including CSPF (constrained shortest path first). MPLS TE-LSPs can be configured within a single domain, e.g., area, level, or AS, or may also span multiple domains, e.g., areas, levels, or ASes." Vasseur '879, 3:7-24.<br><br>"The PCE is an entity having the capability to compute paths between any nodes of which the PCE is aware in an AS or area. PCEs are especially useful in that they are more cognizant of network traffic and path selection within their AS or area, and thus may be used for more optimal path computation. A head-end LSR may further operate as a path computation client (PCC) configured to send a path computation request to the PCE, and receive a response with the computed path, potentially taking into consideration other path computation requests from other PCCs. It is important to note that when one PCE sends a request to another PCE, it acts as a PCC." Vasseur '879, 3:25-36.<br><br>"Some applications may incorporate unidirectional data flows configured to transfer time-sensitive traffic from a source (sender) in a computer network to a destination (receiver) in the network in accordance with a certain "quality of service" (QoS). Here, network resources may be reserved for the unidirectional flow to ensure that the QoS associated with the data flow is maintained. The Resource ReSerVation Protocol (RSVP) is a network-control protocol that |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | enables applications to reserve resources in order to obtain special QoS for their data flows. RSVP works in conjunction with routing protocols to, e.g., reserve resources for a data flow in a computer network in order to establish a level of QoS required by the data flow. RSVP is defined in R. Braden, et al., Resource ReSerVation Protocol (RSVP), RFC 2205, the contents of which are hereby incorporated by reference in its entirety. In the case of traffic engineering applications, RSVP signaling (with Traffic Engineering extensions) is used to establish a TE-LSP and to convey various TE-LSP attributes to routers, such as border routers, along the TE-LSP obeying the set of required constraints whose path may have been computed by various means." Vasseur '879, 3:37-57.<br><br>"Generally, a tunnel is a logical structure that encapsulates a packet (a header and data) of one protocol inside a data field of another protocol packet with a new header. In this manner, the encapsulated data may be transmitted through networks that it would otherwise not be capable of traversing. More importantly, a tunnel creates a transparent virtual network link between two network nodes that is generally unaffected by physical network links or devices (i.e., the physical network links or devices merely forward the encapsulated packet based on the new header). While one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 3:58-4:3.<br><br>"Occasionally, a network element (e.g., a node or link) will fail, causing redirection of the traffic that originally traversed the failed network element to other network elements that bypass the failure. Generally, notice of this failure is relayed to the nodes in the network through an advertisement of the new network topology, e.g., an IGP or BGP Advertisement, and routing tables are updated to avoid the failure accordingly. Reconfiguring a network in response to a network element failure using, e.g., pure IP rerouting, can be time consuming. Many recovery techniques, however, are available to provide fast recovery and/or network configuration in the event of a network element failure, including, inter alia, "Fast Reroute", e.g., MPLS TE Fast Reroute. An example of MPLS TE Fast Reroute is described in Pan, et al., Fast Reroute Extensions to RSVP-TE for LSP Tunnels, RFC 4090, May 2005, which is hereby incorporated by reference as though fully set forth herein." Vasseur '879, 4:4-21. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "Fast Reroute (or FRR) has been widely deployed to protect against network element failures, where "backup tunnels" are created to bypass one or more protected network elements (e.g., links, shared risk link groups (SRLGs), and nodes). When the network element fails, traffic is quickly diverted ("Fast Rerouted") over a backup tunnel to bypass the failed element, or more particularly, in the case of MPLS, a set of primary TE-LSPs (tunnels) is quickly diverted. Specifically, the point of local repair (PLR) node configured to reroute the traffic inserts ("pushes") a new label for the backup tunnel, and the traffic is diverted accordingly. Once the failed element is bypassed, the backup tunnel label is removed ("popped"), and the traffic is routed along the original path according to the next label (e.g., that of the original TE-LSP). Notably, the backup tunnel, in addition to bypassing the failed element along a protected primary TE-LSP, also intersects the primary TE-LSP, i.e., it begins and ends at nodes along the protected primary TE-LSP." Vasseur '879, 4:22-39.<br><br>"To offer maximum protection, e.g., guaranteed bandwidth, during Fast Reroute, backup tunnels may reserve a configurable amount of bandwidth to ensure that Fast Rerouted traffic from the primary tunnel has a reserved path to follow. For example, the bandwidth reserved for the primary tunnel may also be reserved on the backup tunnel. While this approach provides maximum protection, it also requires a non-negligible amount of network resources (e.g., capacity/bandwidth) and may increase operational complexity." Vasseur '879, 4:40-48.<br><br>"Certain techniques are available to efficiently minimize the amount of resources required by the establishment and maintenance of the backup tunnels for Fast Reroute. One such technique is to create zero-bandwidth ("0-BW") backup tunnels (i.e., tunnels that reserve no bandwidth) to protect non-0-BW primary tunnels. This "best effort" approach does not guarantee that the path followed by the backup tunnel will have enough bandwidth to support the diverted primary tunnel at the time of failure without QoS degradation, however in many situations the path quality of the backup tunnel is sufficient. For instance, if the network is not overly congested, or the backup tunnel follows a non-congested path, there may be enough available bandwidth along the backup tunnel to support the newly rerouted traffic. Also, because primary tunnels often reserve bandwidth in response to "peak" traffic utilization, the amount of traffic over the primary tunnel at the time of failure may be far less than the reserved bandwidth (e.g., at "off-peak" times). Notably, those skilled in the art will understand that in |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | the absence of the above exceptions, a 0-BW backup tunnel may have unacceptable bandwidth (e.g., affecting path quality) to support the diverted traffic." Vasseur '879, 4:49-5:2.<br><br>"Currently, head-end nodes (LSRs) may be configured to systematically reroute the primary tunnels affected by the network element failure (e.g., diverted primary tunnels), especially in the case with 0-BW backup tunnels, such as, e.g., by reestablishing a new primary tunnel that follows a path excluding the failed network element. In particular, 0-BW backup tunnels represent a best effort attempt to allow the head-end node to more gracefully reestablish the primary tunnel in response to a failure, since the backup tunnels may not be able to support the diverted traffic without QoS degradation. The systematic reestablishing may potentially result in the reestablishment of a large number of primary tunnels (e.g., up to 3000 for a single network element failure in today's networks). Notably, reestablishing diverted primary tunnels may be undesirable for the network, such as by creating traffic churn, jitter, control plane overloads, etc., as will be understood by those skilled in the art. However, as noted above, there are situations where the backup tunnel may provide acceptable bandwidth, at least, for example, for a period of time (e.g., possibly short) until the failed network element is restored. In these situations, then, it may have been unnecessary to reestablish the diverted primary tunnels. There remains a need, therefore, for a technique that dynamically determines whether to reestablish a diverted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 5:3-28.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 5:32-47. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as Resource ReSerVation Protocol (RSVP) Traffic Engineering (TE) signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or type/length/value (TLV) encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object of the RSVP messages to convey the path quality notification." Vasseur '879, 5:48-58.<br><br>"In accordance with one aspect of the present invention, the head-end node requests the establishment of the primary tunnel (e.g., a TE-Label Switched Path, TE-LSP), along with a request for Fast Reroute protection of one or more network elements (e.g., with zero-bandwidth, 0-BW backup tunnels) at a PLR node. In addition, the head-end node may include a request for backup tunnel path quality notification, such as, e.g., through the use of the novel Feedback sub-object. The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and sends an error message (e.g., an RSVP PathErr) to the head end node indicating the "Fast Rerouting" of the primary tunnel." Vasseur '879, 5:59-6:6.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, the PLR node may collect metrics/statistics (e.g., packet drops, path cost, jitter, etc.) of the primary and/or backup tunnels. Once the primary tunnel is Fast Rerouted, the PLR node continues to collect metrics of the backup tunnel, and may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. Notably, the PLR node may be configured to send path quality notifications to the head-end node once, continually, periodically, in response to configurable changes in path quality, etc. Also, as in the case where multiple primary tunnels are Fast Rerouted, the path quality notification may include an indication of which Fast Rerouted primary tunnels in particular have been effected by the changed path quality." Vasseur '879, 6:7-23. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "In accordance with yet another aspect of the present invention, upon receiving the error message (PathErr), the head-end node may wait for the path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. Notably, in the event the head-end node does not receive a path quality notification for the backup tunnel (e.g., within a configurable time limit), the head-end node may attempt to reestablish the new primary tunnel. Moreover, where the head-end node has multiple primary tunnels being Fast Rerouted, a configurable subset of the primary tunnels may be reestablished, e.g., to reduce congestion of the backup tunnels, and/or to limit the number of reestablished primary tunnels within a given period of time." Vasseur '879, 6:24-43.<br><br>"Advantageously, the novel technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. By providing the head-end node of the primary tunnel with path quality feedback of the backup tunnel, the novel technique avoids reestablishing a potentially large number of tunnels over one or more alternate paths after a failure (and Fast Reroute) if the backup tunnels have acceptable path quality. In particular, the backup tunnels, e.g., 0-BW backup tunnels, may not be congested or subsequently burdened by the Fast Rerouted traffic of the primary tunnel. Also, the failed network element (thus the primary tunnel) may be quickly restored; therefore by not reestablishing the primary tunnel, network jitter, churn, etc., may be avoided. Further, the dynamic nature of the novel technique alleviates the need for cumbersome manual configuration." Vasseur '879, 6:44-59.<br><br>"FIG. 3 is schematic block diagram of an exemplary signaling (RSVP) message that may be advantageously used with the present invention." Vasseur '879, 7:6-8.<br><br>"FIG. 5 is a schematic block diagram of the computer network in FIG. 1 showing Fast Reroute protection of a primary tunnel using a backup tunnel in accordance with the present invention." Vasseur '879, 7:12-15. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 6 is a schematic block diagram of the computer network in FIG. 5 showing an unacceptable backup tunnel path quality and resultant reestablishing of the primary tunnel in accordance with the present invention." Vasseur '879, 7:16-19.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention." Vasseur '879, 7:20-24.<br><br>"FIG. 1 is a schematic block diagram of an exemplary computer network 100 comprising a plurality of nodes A-F, such as routers or other network devices, interconnected as shown. The nodes may be a part of one or more autonomous systems, routing domains, or other networks or subnetworks. For instance, routers A and E may be provider edge (PE) devices of a provider network, (e.g., a service provider network) that are interconnected to one or more customer networks through customer edge (CE) devices (not shown, while the remaining nodes B-D and F may be core provider (P) devices, as will be understood by those skilled in the art. Those skilled in the art will also understand that the nodes A-F may be any nodes within any arrangement of computer networks, and that the view shown herein is merely an example. For example, the nodes may be configured as connections to/from one or more virtual private networks (VPNs), as will be understood by those skilled in the art. These examples are merely representative. Those skilled in the art will understand that any number of routers, nodes, links, etc. may be used in the computer network 100 and connected in a variety of ways, and that the view shown herein is for simplicity." Vasseur '879, 7:29-49.<br><br>"Data packets may be exchanged among the computer network 100 using predefined network communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Asynchronous Transfer Mode (ATM) protocol, Frame Relay protocol, Internet Packet Exchange (IPX) protocol, etc. Routing information may be distributed among the routers of the computer network using predetermined Interior Gateway Protocols (IGPs), such as conventional distance-vector protocols or, illustratively, link-state protocols, through the use of IGP Advertisements." Vasseur '879, 7:50-60. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 2 is a schematic block diagram of an exemplary router 200 that may be advantageously used with the present invention, e.g., as an edge router or a core router. The router comprises a plurality of network interfaces 210, a processor 220, and a memory 240 interconnected by a system bus 250. The network interfaces 210 contain the mechanical, electrical and signaling circuitry for communicating data over physical links coupled to the network 100. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols, including, inter alia, TCP/IP, UDP, ATM, synchronous optical networks (SONET), wireless protocols, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), etc." Vasseur '879, 7:61-8:6.<br><br>"The memory 240 comprises a plurality of storage locations that are addressable by the processor 220 and the network interfaces 210 for storing software programs and data structures associated with the present invention. The processor 220 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures. A router operating system 242 (e.g., the Internetworking Operating System, or IOS™, of Cisco Systems, Inc.), portions of which is typically resident in memory 240 and executed by the processor, functionally organizes the router by, inter alia, invoking network operations in support of software processes and/or services executing on the router. These software processes and/or services may comprise routing services 247, Traffic Engineering (TE) services 244, and RSVP services 249. It will be apparent to those skilled in the art that other processor and memory means, including various computer-readable media, may be used to store and execute program instructions pertaining to the inventive technique described herein." Vasseur '879, 8:7-26.<br><br>"Routing services 247 contain computer executable instructions executed by processor 220 to perform functions provided by one or more routing protocols, such as IGP (e.g., OSPF and IS-IS), IP, BGP, etc. These functions may be configured to manage a forwarding information database (not shown) containing, e.g., data used to make forwarding decisions. Routing services 247 may also perform functions related to virtual routing protocols, such as maintaining VRF instances (not shown) as will be understood by those skilled in the art." Vasseur '879, 8:27-36. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "RSVP services 249 contain computer executable instructions for implementing RSVP and processing RSVP messages in accordance with the present invention. RSVP is described in RFC 2205, entitled Resource ReSerVation Protocol (RSVP), and in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels, both as incorporated above." Vasseur '879, 8:37-42.<br><br>"TE services 244 contain computer executable instructions for operating TE functions in accordance with the present invention. Examples of Traffic Engineering are described in RFC 3209, RFC 3784, and RFC 3630 as incorporated above, and in RFC 3473, entitled, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions dated January 2003, which is hereby incorporated by reference in its entirety. A TE database (TED, not shown) may be illustratively resident in memory 240 and used to store TE information provided by the routing protocols, such as IGP, BGP, and/or RSVP (with TE extensions, e.g., as described herein), including, inter alia, path quality information as described herein. The TED may be illustratively maintained and managed by TE services 244." Vasseur '879, 8:43-57.<br><br>"Changes in the network topology may be communicated among routers 200 using a link-state protocol, such as the conventional OSPF and IS-IS protocols. Suppose, for example, that a communication link fails or a cost value associated with a network node changes. Once the change in the network's state is detected by one of the routers, that router may flood an IGP Advertisement communicating the change to the other routers in the network. In this manner, each of the routers eventually "converges" to an identical view of the network topology." Vasseur '879, 8:58-67.<br><br>"In one embodiment, the routers described herein are IP routers that implement Multi-Protocol Label Switching (MPLS) and operate as label switched routers (LSRs). In one simple MPLS scenario, at an ingress to a network, a label is assigned to each incoming packet based on its forwarding equivalence class before forwarding the packet to a next-hop router. At each router, a forwarding selection and a new substitute label are determined by using the label found in the incoming packet as a reference to a label forwarding table that includes this information. At the network egress, a forwarding decision is made based on the incoming label |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | but optionally no label is included when the packet is sent on to the next hop. In some network configurations, one hop prior to the network egress, a penultimate hop popping (PHP) operation may be performed. Particularly, because the hop prior to the network egress (the penultimate hop) is attached to the network egress, the label is no longer needed to assure that the traffic follows a particular path to the network egress. As such, the PHP-enabled device "pops" the labels from the traffic before forwarding the traffic to the network egress, e.g., using conventional or native (IP) routing, thereby alleviating the task of removing the labels at the network egress." Vasseur '879, 9:1-23.<br><br>"The paths taken by packets that traverse the network in this manner are referred to as label switched paths (LSPs) or Traffic Engineering (TE)-LSPs. An example TE-LSP is shown as the thick line and arrow (T1) between a head-end node (router A) and a tailend node (router E) in FIG. 1. Establishment of a TE-LSP requires computation of a path, signaling along the path, and modification of forwarding tables along the path. MPLS TE establishes LSPs that have guaranteed bandwidth under certain conditions. Illustratively, the TE-LSPs may be signaled through the use of the RSVP protocol (with Traffic Engineering extensions), and in particular, RSVP TE signaling messages. Notably, when incorporating the use of PCEs (described below), the path computation request (and response) between PCC and PCE can be exchanged in accordance with a protocol specified in Vasseur, et al., Path Computation Element (PCE) Communication Protocol (PCEP)—Version 1—<draft-vasseur-pce-pcep-02.txt>, Internet Draft, September 2005, the contents of which are hereby incorporated by reference in its entirety. It should be understood that the use of RSVP or PCEP serves only as an example, and that other communication protocols may be used in accordance with the present invention." Vasseur '879, 9:24-45.<br><br>"In accordance with RSVP, to request a data flow (TE-LSP) between a sender and a receiver, the sender may send an RSVP path request (Path) message downstream to the receiver along a path (e.g., a unicast route) to identify the sender and indicate e.g., bandwidth needed to accommodate the data flow, along with other attributes of the TE-LSP. The Path message may contain various information about the data flow including, e.g., traffic characteristics of the data flow. Also in accordance with the RSVP, a receiver establishes the TE-LSP between the sender and receiver by responding to the sender's Path message with a reservation request |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | (Resv) message. The reservation request message travels upstream hop-by-hop along the flow from the receiver to the sender. The reservation request message contains information that is used by intermediate nodes along the flow to reserve resources for the data flow between the sender and the receiver, to confirm the attributes of the TE-LSP, and provide a TE-LSP label. If an intermediate node in the path between the sender and receiver acquires a Path message or Resv message for a new or established reservation (TE-LSP) and encounters an error (e.g., insufficient resources, failed network element, etc.), the intermediate node generates and forwards a path or reservation error (PathErr or ResvErr, hereinafter Error) message to the sender or receiver, respectively." Vasseur '879, 9:46-10:2.<br><br>"FIG. 3 is a schematic block diagram of portions of a signaling message 300 (e.g., RSVP message, such as Path, Resv or Error) that may be advantageously used with the present invention. Message 300 contains, inter alia, a common header 310 and one or more signaling protocol specific objects 320, such as an LSP-ATTRIBUTE object 330. The common header 310 may comprise a source address 312 and destination address 314, denoting the origination and requested termination of the message 300. Protocol specific objects 320 contain objects necessary for each type of message 300 (e.g., Path, Resv, Error, etc.). For instance, a Path message may have a sender template object, Tspec object, Previous-hop object, etc. The LSP-ATTRIBUTE object 330, for instance, may be used to signal attributes and/or information regarding an LSP (tunnel). To communicate this information, LSP-ATTRIBUTE object 330 (as well as specific objects 320) may include various type/length/value (TLV) encoding formats and/or flags, as will be understood by those skilled in the art. An example of an LSP-ATTRIBUTE object is further described in Farrel, et al., Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE <draft-ietf-mpls-rsvpte-attributes-05.txt>, Internet Draft, May 2005, which is hereby incorporated by reference as though fully set forth herein. A Resv message, on the other hand, may have specific objects 320 for a label object, session object, filter spec object, etc., in addition to the LSP-ATTRIBUTE object 330. Error messages 300 (e.g., PathErr or ResvErr), may also have specific objects 320, such as for defining the type of error, etc." Vasseur '879, 10:3-31. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "It should be noted that in accordance with RSVP signaling, the state of the TE-LSP is refreshed on a timed interval, e.g., every thirty seconds, in which RSVP Path and Resv messages are exchanged. This timed interval is configurable by a system administrator. Moreover, various methods understood by those skilled in the art may be utilized to protect against route record objects (RROs) contained in signaling messages for a TE-LSP in the event security/privacy is desired. Such RRO filtering prevents a head-end node of the TE-LSP from learning of the nodes along the TE-LSP, i.e., nodes within the provider network." Vasseur '879, 10:4-42.<br><br>"Although the illustrative embodiment described herein is directed to MPLS, it should also be noted that the present invention may advantageously apply to Generalized MPLS (GMPLS), which pertains not only to packet and cell-based networks, but also to Time Division Multiplexed (TDM) and optical networks. GMPLS is well known and described in RFC 3945, entitled Generalized Multi-Protocol Label Switching (GMPLS) Architecture, dated October 2004, and RFC 3946, entitled Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control, dated October 2004, the contents of both of which are hereby incorporated by reference in their entirety." Vasseur '879, 10:43-55.<br><br>"To obviate delays associated with updating routing tables when attempting to avoid a failed network element (i.e., during convergence), some networks have employed MPLS TE Fast Reroute (FRR). MPLS Fast Reroute is a technique that may be used to quickly divert ("Fast Reroute") traffic around failed network elements in a TE-LSP. MPLS Fast Reroute is further described, for example, by Fast Reroute Extensions to RSVP-TE for LSP Tunnels, as incorporated by reference above. According to the technique, one or more network elements (e.g. links or nodes) in a network are protected by backup tunnels following an alternate path. If a failure occurs on a protected link or node, TE-LSPs (and consequently the traffic that they carry) are locally diverted onto an appropriate alternate path (e.g., a "backup tunnel") by the node immediately upstream from the failure. The backup tunnel acts as a Fast Reroute path for the primary TE-LSP and obviates delays associated with other measures, such as tearing down the primary TE-LSP after having gracefully diverted the TE-LSPs affected by the failure, should an alternate path around the failed network element exist. In the event of a |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | failure of a protected element the head-end node of the backup tunnel (or a "point of local repair," PLR node) may quickly begin diverting traffic over the backup tunnel with minimal disruption to traffic flow. Those skilled in the art will understand that MPLS Fast Reroute is one example of link or node failure protection, and that other known correction mechanisms may be used in accordance with the present invention. As mentioned above, however, the head-end node of the Fast Rerouted primary tunnel may attempt to reestablish the primary tunnel in response to learning of the protected element failure, particularly in the case where the backup tunnel is a zero-bandwidth (0-BW) tunnel. The attempt to reestablish the primary tunnel has conventionally been a systematic response to Fast Rerouting (diverting) of the primary tunnel, regardless of the path quality of the backup tunnel." Vasseur '879, 10:56-11:23.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 11:24-39.<br><br>"In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as RSVP TE signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or TLV encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object 330 of the RSVP messages 300 to convey the path quality notification." Vasseur '879, 11:40-49. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 4 is a schematic block diagram illustrating the format of an extension object (TLV) 400, such as a Fast Reroute Feedback object, that may be advantageously used with the present invention. The extension object (or sub-object) 400 is illustratively embodied as a TLV contained in an LSP-ATTRIBUTE object 330 of an RSVP message 300 and is extended to carry backup (and/or primary) tunnel path quality information. To that end, the "Feedback" object 400 is organized to include a Type field 405 containing a predetermined type value signifying the specific content of the object 400. The Length field 410 is a variable length value. The TLV encoded format may also comprise one or more non-ordered sub-TLVs 450 carried within the TLV "payload" (e.g. Value field 415), each having a Type field 455, Length field 460, and Value field 465. The fields of the TLV 400 and sub-TLV(s) 450 are used in a variety of manners, including as described herein, according to the present invention." Vasseur '879, 11:59-12:8.<br><br>"In accordance with one aspect of the present invention, the head-end node (e.g., router A) requests the establishment of the primary tunnel (e.g., T1), such as a TE-LSP. Notably, the head-end node may be a head-end node for multiple primary tunnels, as will be understood by those skilled in the art. Along with the primary tunnel establishment, the head-end node may also request Fast Reroute protection of one or more network elements (e.g., all intermediate network elements) at a PLR node (e.g., router B as shown). Note that each intermediate node along the primary tunnel may act as a PLR node, and that router B is shown merely for simplicity. Illustratively, the Fast Reroute protection may be embodied as one or more zero-bandwidth (0-BW) backup tunnels at the PLR node (e.g., BT1). Those skilled in the art will also understand that the PLR node may protect more than one primary tunnel originating at more than one corresponding head-end node (not shown)." Vasseur '879, 12:9-25.<br><br>"The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. For example, various connectivity verification protocols, such as, e.g., Bidirectional Forwarding Detection (BFD), IGP "Hello" packets, BGP KEEPALIVE messages, etc., may be used to detect a failure of a network element, as will be understood by those skilled in the art. Furthermore, other lower-layer failure detection mechanisms (e.g. optical or SONET/SDH alarms) may be used to detect a network element failure. Once failure is detected, the PLR node diverts the |

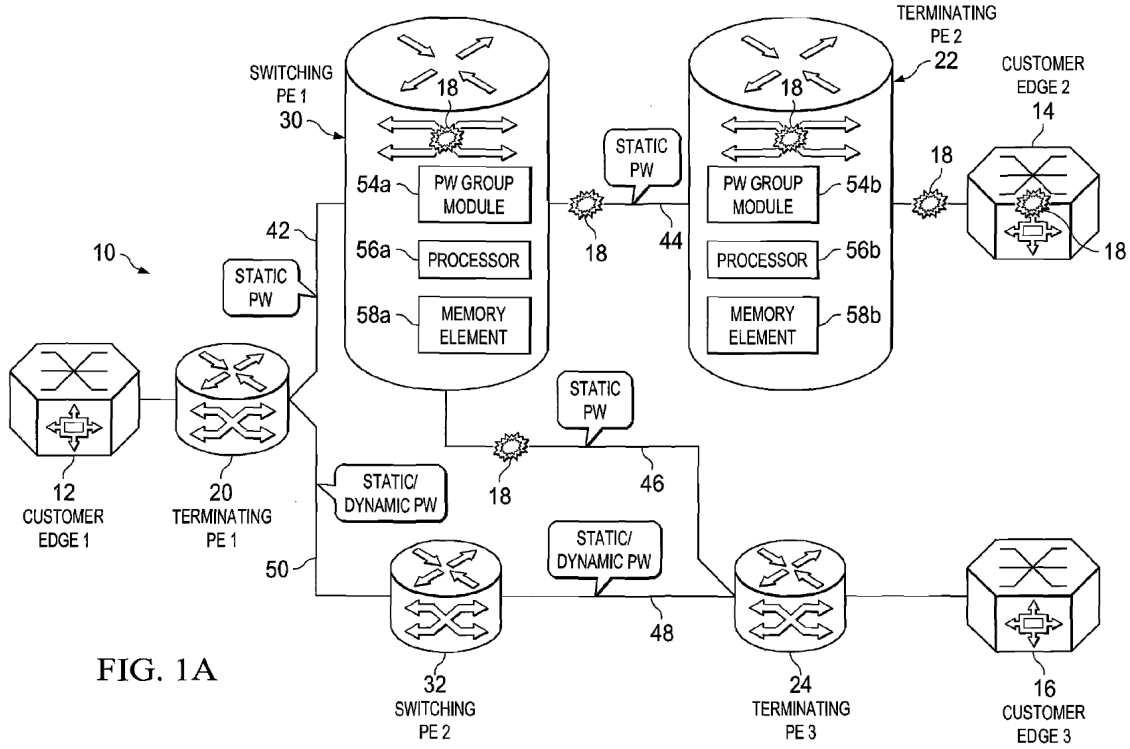
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | traffic onto the backup tunnel, and may send an error message (e.g., an RSVP PathErr 300, such as a conventional "tunnel locally repaired" message) to the head end node indicating the "Fast Rerouting" of the primary tunnel. FIG. 5 is a schematic block diagram of the computer network 100 in FIG. 1 showing Fast Reroute protection of the primary tunnel T1 (e.g., in response to a protected network element, router C, failure, indicated with an overlaid "X") using a backup tunnel BT1 in accordance with the present invention. Traffic originally received at the PLR node (router B) over the primary tunnel is now diverted over the backup tunnel to a remerge point (router D) of the primary tunnel, as will be understood by those skilled in the art." Vasseur '879, 12:42-65.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, each PLR node may collect metrics/statistics of the primary and/or backup tunnels. For instance, example metrics may comprise, inter alia, packet drops, path cost, jitter, delay, bandwidth, etc. The PLR node may collect the metrics through traffic monitoring, probes, independent calculations, and/or through cooperation with protected nodes of the primary tunnel (primary nodes) and nodes of the backup tunnel (backup nodes), e.g., transmitting path quality notifications. Once the primary tunnel is Fast Rerouted (i.e., diverted after failure of a protected network element), the PLR node continues to collect metrics of the backup tunnel. (Alternatively, metrics of the backup tunnel may be collected only after Fast Reroute, and not prior to Fast Reroute.)" Vasseur '879, 12:66-13:12.<br><br>"For example, a path cost increase from the primary tunnel to the backup tunnel may be computed (and transmitted, below) by the PLR node prior to the failure (or during the failure while the primary TE-LSP is diverted onto the backup tunnel) using its own routing tables. The path cost increase may be calculated as a difference between the entire length (head-end node to tail-end node) of the primary and backup tunnels, or just the difference between the protected segment of the primary tunnel and the backup tunnel (PLR node to remerge point). Also, a jitter increase of the primary and backup tunnels, which may be generally described as a difference between inter-arrival of packets, may be monitored using various known techniques, such as, e.g., sending probe packets (probes) from the PLR node to the remerge point. For instance, probes may determine that packets arrive at the remerge point along the primary tunnel from the PLR node consistently, e.g., every 10 milliseconds (ms) (e.g., an |

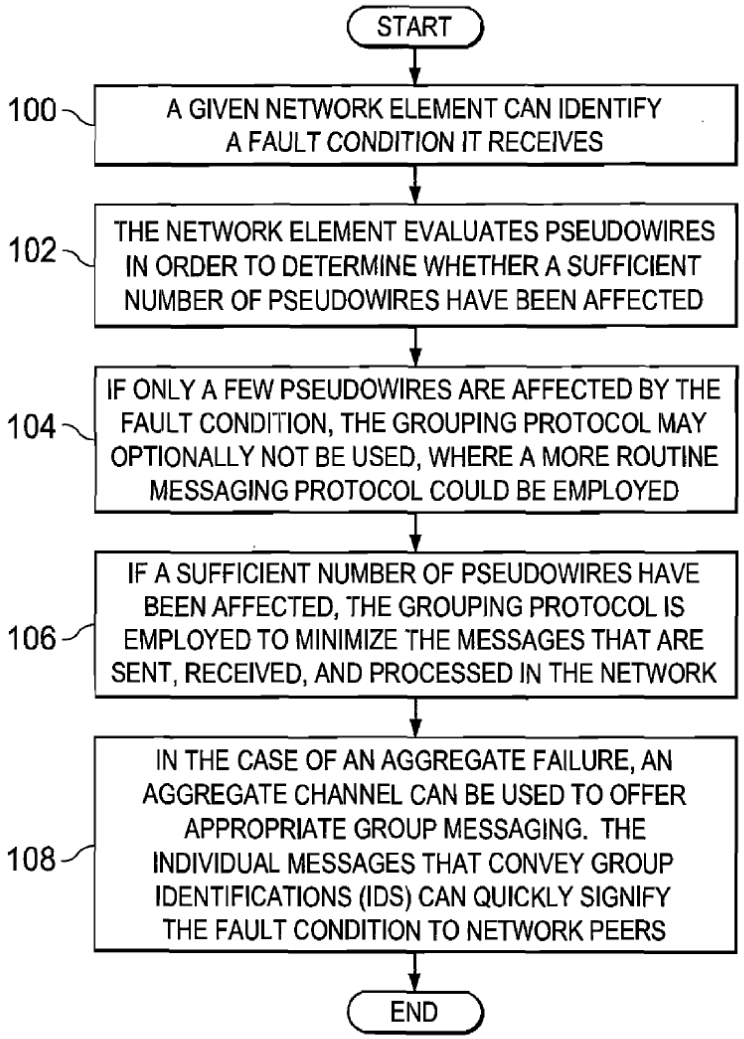
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | average value). After Fast Reroute, however, probes may determine that packets do not arrive at the remerge point along the backup tunnel from the PLR node consistently, e.g., one may arrive in 10 ms, another in 100 ms, another in 50 ms, etc. The non-constant rate of received packets (jitter) may be undesirable, e.g., in particular for voice over IP (VoIP) traffic, as will be understood by those skilled in the art." Vasseur '879, 13:13-36.<br><br>"As a further example, packet dropping may be monitored for the primary and backup tunnels prior to and after Fast Reroute (respectively). For instance, based on the tunnel label of the dropped packet, primary nodes and/or backup nodes may be able to distinguish which tunnel corresponds to the dropped packets. Each of the primary and/or backup nodes collect packet drop data, and periodically inform the PLR node of the number of dropped packets (e.g., though a corresponding Feedback object 400). In the case of a backup node, the PLR node receiving the notification may interpret the association of the backup tunnel label and the primary tunnel label to reference an appropriate primary tunnel. Those skilled in the art will understand that the above path quality metrics are merely examples, and that any other metrics/statistics useful for determining path quality of the backup tunnel may be used in accordance with the present invention (e.g., delay, bandwidth, etc.). Further, the path quality information may be measured and compared in a variety of manners, such as, e.g., as a difference between primary and backup tunnels before and after Fast Reroute, or simply the difference between the backup tunnel before and after Fast Reroute, etc." Vasseur '879, 13:37-58.<br><br>"Also after the primary tunnel is Fast Rerouted, the PLR node may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. For instance, the novel Feedback object 400 may include one or more sub-TLVs 450 corresponding to metrics/statistics, as described above. Notably, the path quality information pertaining to a particular metric/statistic may be transmitted as total values for interpretation by the head-end node (e.g., to determine the difference), or as PLR-node-computed differences (e.g., between the primary and backup tunnels, or before and after Fast Reroute). For instance, if the delay of the primary tunnel (along the protected segment) prior to Fast Reroute were 2 ms, and after Fast Reroute the delay of the backup tunnel were 5 ms, the notification may be configured to |

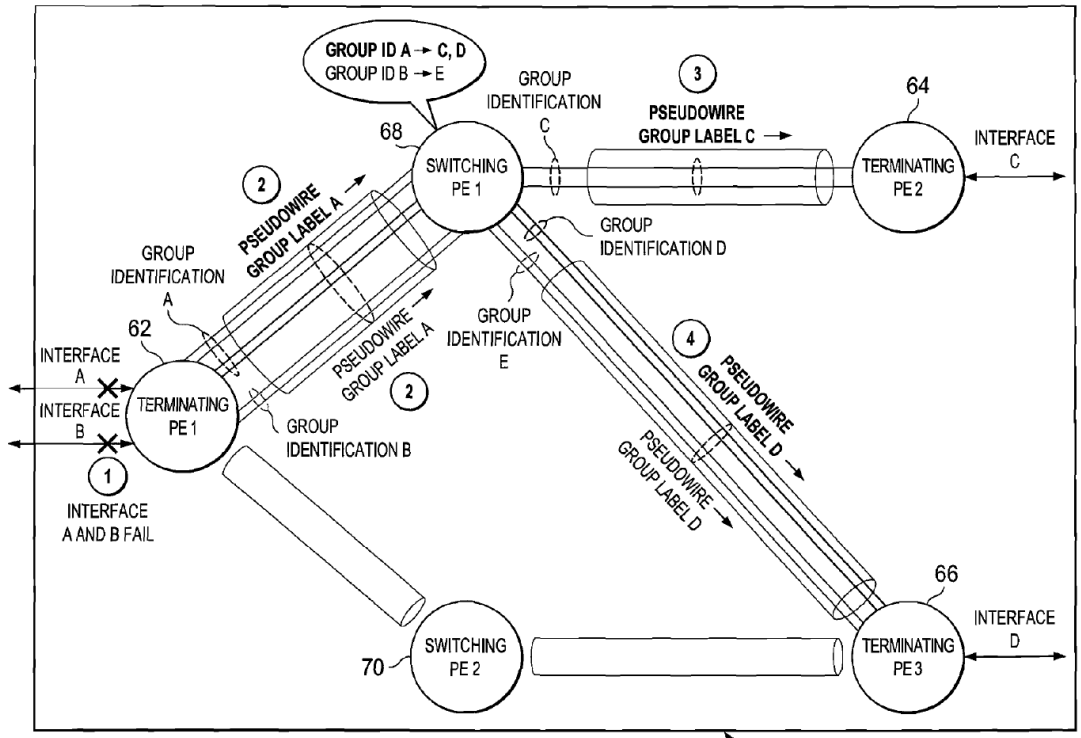
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | include both values 2 ms and 5 ms, or instead simply the difference, i.e., an increase of 3 ms." Vasseur '879, 13:59-14:8.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message 300 (PathErr), the head-end node may wait for at least one path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. Also, any number of metrics may be used in the determination, e.g., as configured by a system administrator. For example, using the metrics described above, a head-end node may be configured to reestablish the primary tunnel in response to i) a certain number of packet drops, ii) a percent increase in packet drops, iii) a number of packet drops and a percent increase in path cost, iv) a percent increase in path cost and a percent increase in jitter, etc. Those skilled in the art will understand that these are merely examples of possible path quality comparisons and reestablishment determinations, and that any comparisons to any metrics at any configurable changes may be used in accordance with the present invention." Vasseur '879, 14:41-59.<br><br>"If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. FIG. 6 is a schematic block diagram of the computer network 100 in FIG. 5 showing an unacceptable backup tunnel path quality (dotted line and arrow) and resultant reestablishment of the primary tunnel in accordance with the present invention. Those skilled in the art will understand that the reestablished primary tunnel may traverse one or more primary nodes (not shown), and that it may be computed specifically to avoid the failed network element and any network elements of the unacceptable backup tunnel. Those skilled in the art will also understand that the attempt to reestablish the primary tunnel may not be able to find an acceptable path, in which case the head-end node may continue to use the unacceptable backup tunnel or other unacceptable rerouted path." Vasseur '879, 14:60-15:9. |

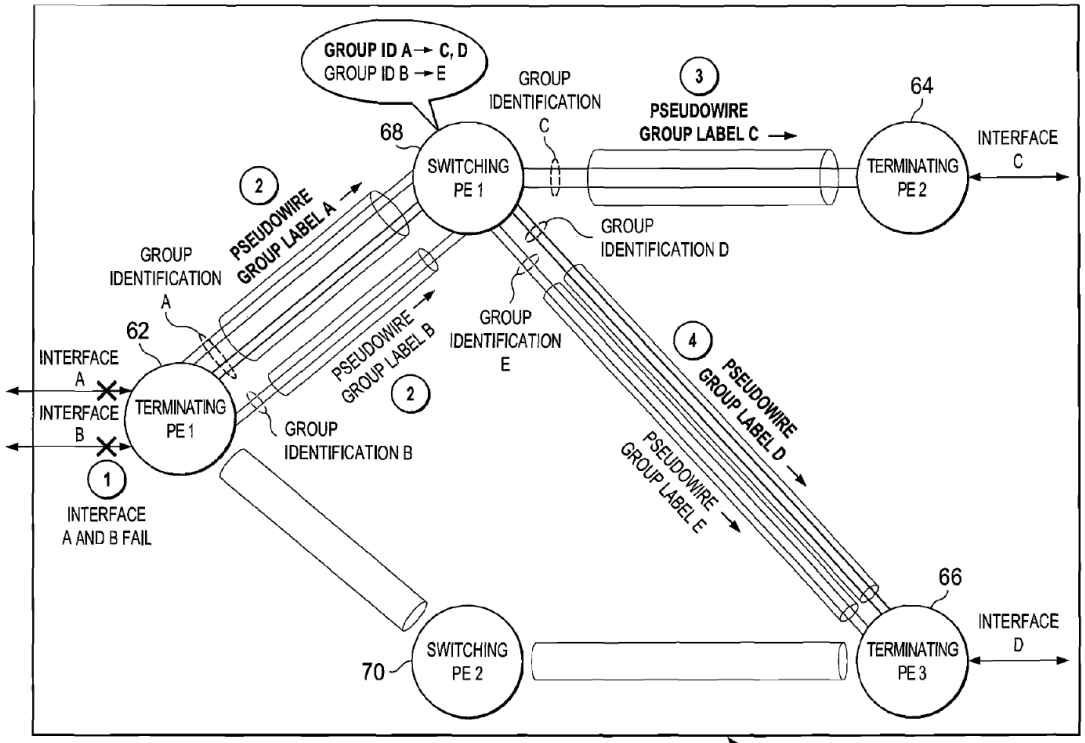
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention. The procedure 700 starts at step 705, and continues to step 710, where a head-end node (e.g., router A) establishes a primary tunnel (e.g., T1) to a destination tail-end node (e.g., router E). During or after establishment, the head-end node requests protection of the primary tunnel at step 715, and in response, PLR nodes along the primary tunnel (e.g., router B) create backup tunnels (e.g., BT1) to protect the primary tunnel in step 720. (Those skilled in the art will understand that backup tunnels around the protected network elements may already exist at the PLR node, and that "creating backup tunnels" in step 720 implies an association with pre-existing backup tunnels.) As mentioned above, these backup tunnels may illustratively be embodied as 0-BW backup tunnels. In accordance with the present invention, the head-end node may additionally request backup tunnel path quality notification from the PLR nodes in step 725, such as, e.g., through the use of empty corresponding Feedback objects in RSVP (Path) messages 300, as described above." Vasseur '879, 15:37-58.<br><br>"The procedure 700 continues to FIG. 7B (step "A"), where in step 750 the PLR node detecting the failure diverts ("Fast Reroutes") the primary tunnel traffic to the backup tunnel and sends an error message (e.g., an RSVP (Error) message 300) to the head end node, e.g., a "tunnel locally repaired" message. The detecting PLR node continues to collect path quality information for the backup tunnel in step 755 and at step 760 determines whether to send the path quality notification to the head-end node. For example, as mentioned above, the PLR node may be configured to continually send notifications, or periodically, or in response to a configurable change in path quality, etc. Also as mentioned above, the PLR node may be configured to send either the actual path quality information or the change (difference) in path quality. (As further mentioned above, metrics of the backup tunnel may alternatively be collected only after Fast Reroute, and not prior to Fast Reroute.)." Vasseur '879, 16:4-20.<br><br>"If the PLR node decides to send the notification in step 760, then the head-end node determines whether to reestablish the primary tunnel based on the backup tunnel path quality notification in step 765, e.g., based on one or more configurable thresholds, percentages, etc., as described above. If the backup tunnel is currently maintaining an acceptable quality for the |

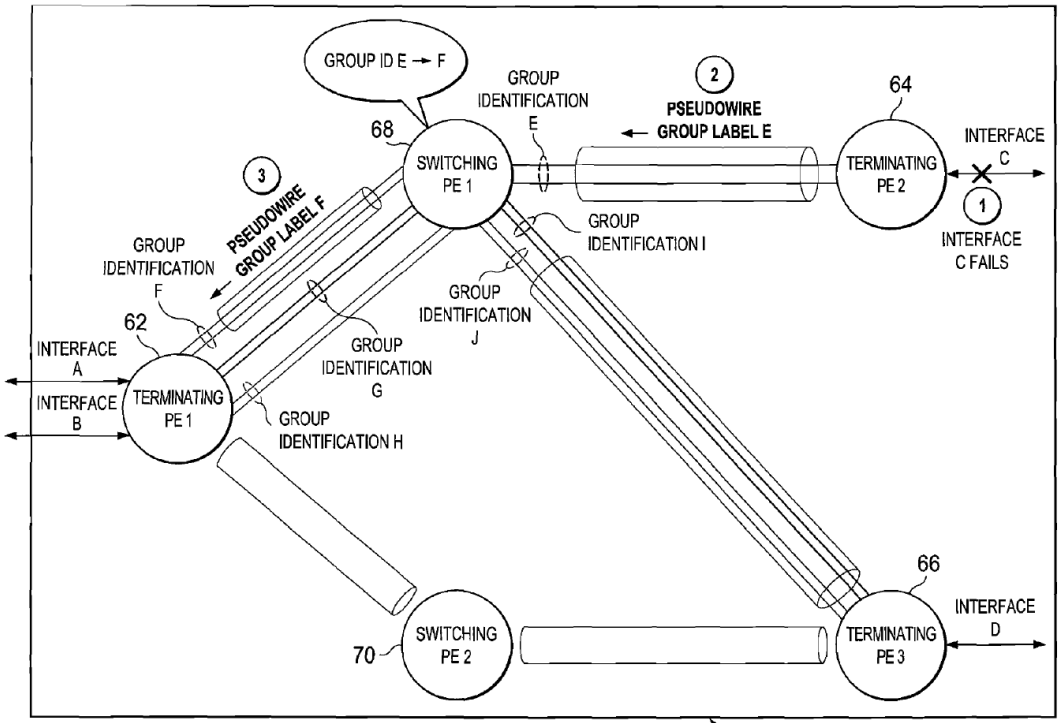
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | traffic flow in step 770, the head-end node may continue to utilize the backup tunnel, and the PLR node continues to collect path quality information in step 755 to detect any change in quality. Otherwise, if the backup tunnel quality is not acceptable in step 770, the head-end node may attempt to reestablish the primary tunnel in step 775. Notably, as mentioned above, if the PLR node has not sent any notification (step 760) within a configurable period of time in step 780, e.g., due to a backup tunnel failure, over-congestion, etc., then the head-end node may also attempt to reestablish the primary tunnel in step 775 accordingly. Moreover, as described above, in the event more than one primary tunnel is Fast Rerouted for the head-end node, various techniques to reestablish one or more of the primary tunnels may be used (e.g., as many tunnels as necessary, a configurable subset of tunnels, all tunnels, the congested tunnels, etc.). The procedure 700 ends in step 785." Vasseur '879, 16:21-43.<br><br>"While there has been shown and described an illustrative embodiment that dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the present invention. For example, the invention has been shown and described herein using "Fast Reroute," e.g., MPLS TE Fast Reroute (FRR). However, the invention in its broader sense is not so limited, and may, in fact, be used with other network element protection and failure correction mechanisms as will be understood by those skilled in the art. Moreover, while the above description describes performing the technique at the head-end node and PLR node, the invention may also be advantageously used with PCEs. In addition, while one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 16:63-17:13.<br><br>**Rustogi discloses**:<br>"An example method includes identifying a fault condition in a network, and evaluating pseudowires affected by the fault condition in order to make a determination as to whether an aggregate failure occurred in the network for a group of pseudowires. The method also includes communicating a group message indicating that the group of pseudowires is associated with the fault condition. The group message includes a group identification (ID), |

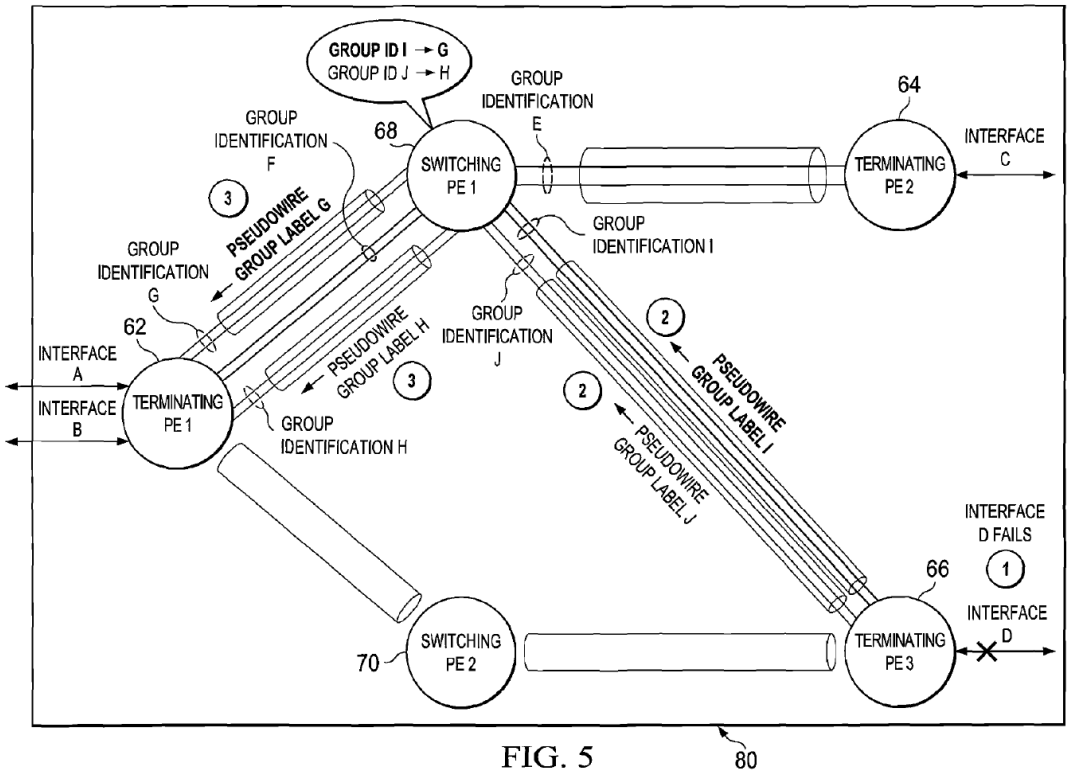
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | which identifies the group of pseudowires, and the group message includes a pseudowire group label identifying an in-band aggregate channel. More specifically, the pseudowire group label can be applicable to static pseudowires. In more detailed embodiments, the group ID identifies the group of pseudowires that are associated with an attachment circuit, a label switched path, or a port. Internal mappings can be maintained such that a plurality of pseudowires is mapped to individual interfaces of network elements in the network." Rustogi, Abstract.<br>  <br><br>FIG. 1A<br>Rustogi, FIG. 1A. |

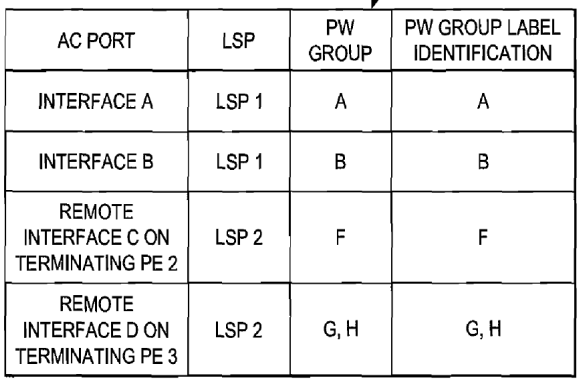
| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 1B<br>Rustogi, FIG. 1B. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Rustogi, FIG. 2. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Rustogi, FIG. 3. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Rustogi, FIG. 4. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 5 80<br>Rustogi, FIG. 5. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | 74<br>  <br>FIG. 6<br>Rustogi, FIG. 6.<br><br>"The field of communications has become increasingly important in today's society. In particular, the ability to quickly and to effectively provision connections presents a significant challenge to component manufacturers, system designers, and network operators. Multiprotocol Label Switching (MPLS) is a mechanism in telecommunications networks that carries data from one network node to the next. Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be emulated over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDUs) and transmitting them over pseudowires. Protocols exist for establishing and maintaining the pseudowires. Certain issues have arisen in pseudowire scenarios, where faults are detected in the network." Rustogi, ¶ [0002].<br><br>"FIG. 1A is a simplified block diagram of a communication system for providing pseudowire group labels in a network environment in accordance with one embodiment of the present disclosure." Rustogi, ¶ [0004]. |

| AC PORT | LSP | PW GROUP | PW GROUP LABEL IDENTIFICATION |
|---|---|---|---|
| INTERFACE A | LSP 1 | A | A |
| INTERFACE B | LSP 1 | B | B |
| REMOTE INTERFACE C ON TERMINATING PE 2 | LSP 2 | F | F |
| REMOTE INTERFACE D ON TERMINATING PE 3 | LSP 2 | G, H | G, H |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 1B is a simplified flowchart depicting one possible, generic operational flow associated with the communication system." Rustogi, ¶ [0005].<br><br>"FIG. 2 is a simplified block diagram of an example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0006].<br><br>"FIG. 3 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0007].<br><br>"FIG. 4 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0008].<br><br>"FIG. 5 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0009].<br><br>"FIG. 6 is a simplified table of an example set of pseudowire group provisioning parameters in accordance with one embodiment." Rustogi, ¶ [0010].<br><br>"FIG. 1A is a simplified block diagram of a communication system 10 for providing pseudowire group labels in accordance with one example of the present disclosure. FIG. 1A includes a customer edge 1 (CE1) 12, a CE2 14, and a CE3 16, where a number of faults 18 are shown as propagating in the network. Typically, when an error or a failure occurs in the network (e.g., an interface failure, a pulled cable, a switch failure, hardware/software failures generally, etc.), messages are sent to various network devices in order to inform them of these fault conditions. Faults 18 of FIG. 1A are indicative of such messages, where the underlying fault condition (being signaled by the messages) can occur virtually anywhere in a network (e.g., in a customer edge, in provider equipment, etc.). FIG. 1A also includes a terminating provider equipment 1 (TPE1) 20, a TPE2 22, a TPE3 24, a switching provider edge 1 (SPE1) 30, and a SPE2 32. In one particular example implementation, the TPEs and SPEs of FIG. 1A are switches that are configured to exchange data in a network environment." Rustogi, ¶ [0012]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "SPE1 30 may include a pseudowire (PW) group module 54 a, a processor 56 a, and a memory element 58 a. In a similar fashion, TPE2 22 may include a pseudowire group module 54 b, a processor 56 b, and a memory element 58 b. FIG. 1A also includes a number of static pseudowires 42, 44, and 46. A set of static/dynamic pseudowires 48, 50 is also provided. Note that the group labeling protocol discussed herein can be executed between individual SPEs, TPEs, or between any combinations of these elements." Rustogi, ¶ [0013].<br><br>"In one particular arrangement, communication system 10 is provided in conjunction with a Layer-2 virtual private networks (L2VPN)/operation, administration, and maintenance (OAM) L2VPN/OAM framework. The OAM framework is intended to provide OAM layering across L2VPN services, pseudowires, and packet switched network (PSN) tunnels. Communication system 10 may include any suitable networking protocol or arrangement that provides a communicative platform for communication system 10. Thus, communication system 10 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Communication system 10 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs." Rustogi, ¶ [0014].<br><br>"Failure detection and failure notification for static pseudowires is inadequate, where sluggish signaling can result in poor scalability for failover performance. Typically, static pseudowires are manually configured at respective endpoints, where control channels are absent for providing group level signaling messages. Aggregate channels are significant tools for providing suitable scalability in the network, but no such aggregate channel exists for static pseudowires. For dynamic pseudowires, such an aggregate channel may be present in the form of a label distribution protocol (LDP) directed session. However, no such protocol exists for static pseudowire configurations such that an in-band aggregate channel would be available for static pseudowires." Rustogi, ¶ [0016]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "Communication system 10 can address the aforementioned issues (and others) by offering a pseudowire group label that can represent an aggregate channel for groups of static pseudowires. The aggregate channel of communication system 10 can allow for improved scalability of failover performance. In accordance with one potential configuration of communication system 10, a pseudowire group label is representative of a group of static pseudowires transported over a label switched path (LSP). The pseudowire group label can identify the aggregate channel, which captures the hierarchy relevant to OAM mechanisms. Additionally, the groups represented by the group identification (ID) can be mutually exclusive, where a pseudowire is part of only one group. In other embodiments, a pseudowire can be part of multiple groups, or be configured in any other suitable manner based on particular network arrangements." Rustogi, ¶ [0017].<br><br>"During operations, and with brief reference to FIG. 1B, a given network element can identify a fault condition it receives (at step 100) and, subsequently, evaluate pseudowires in order to determine whether a sufficient number of pseudowires have been affected. This is reflected by step 102. If only a few pseudowires are affected by the fault condition, the grouping protocol outlined herein may have only nominal value, where there could be an option to simply communicate the fault condition in a more routine manner, as outlined in step 104. However, if a sufficient number of pseudowires have been affected, the grouping protocol outlined herein can be employed to minimize the messages that are sent, received, and processed in the network. This is reflected as step 106. Note that the determination (as to whether a sufficient number of pseudowires have been impacted by the fault condition) can involve accessing internal tables such that a quick mapping can occur to determine if an aggregate failure has occurred. As used herein, the term 'aggregate failure' simply connotes that a sufficient number of pseudowires have experienced the fault condition such that an aggregate channel can be employed to offer appropriate group messaging. For the aggregate failure condition, the individual messages that convey Group identifications (IDs) can quickly signify the fault condition to network peers, as shown in step 108." Rustogi, ¶ [0018].<br><br>"In specific regards to OAM mechanisms, OAM messages typically result from common failures in the network. These fault conditions can be aggregated such that they are signaled as a single message, which could represent a group of failed pseudowires (as opposed to |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | sending individual messages for each failed pseudowire). Hence, a single message could be sent to represent all the relevant OAM messages propagating in communication system 10. The group label that propagates in communication system 10 provides an architecture with a significant level of aggregation for failed pseudowires (i.e., pseudowires being affected by a given fault condition), particularly for OAM messaging. Moreover, the in-band aggregate channel of communication system 10 is based (at least in part) on the evolving trends of OAM mechanisms, which are required to be fast, responsive, and capable of being implemented in hardware or software. Additionally, in-band OAM protocols are a better measure of the path availability." Rustogi, ¶ [0020].<br><br>"In operation of one example implementation, a group label can represent the tuple <attachment circuit (AC) port level grouping, LSP>. This could signify that all pseudowires on an AC port (sought for aggregation) traverse a given LSP. Multiple pseudowire groups can exist within an LSP. Similarly, pseudowires on the same AC port (that traverse a different LSP) can use a different pseudowire group label. Alternatively, an administrator may seek to employ a one-to-one mapping between an LSP and a group label. If that were the case, then only one pseudowire group would exist within an LSP. In scenarios where there is no LSP label in the packet (e.g., due to penultimate hop popping), the pseudowire group label can provide the hierarchy that is appropriate." Rustogi, ¶ [0021].<br><br>"In one particular example, the group level pseudowire OAM message can be sent with the following label stack: Explicit/Implicit LSP Label+pseudowire group Label+GAL+ACH+pseudowire OAM with grouping TLV (where GAL=Generic Associated Channel Label, ACH=Associated Channel Header, TLV=Type-Length-Value). If there are multiple LSPs, then one group label can be provisioned for each LSP (for each pseudowire group), where per group messages can be sent on each LSP. The group label does not necessarily have a one-to-one mapping to the grouping of pseudowires implied by the Group ID in the grouping TLV. Note also that the group-based aggregate channel is applicable to static pseudowires, as well as for dynamic pseudowires in certain applications." Rustogi, ¶ [0022]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "As discussed herein, the aggregate channel of communication system 10 can be configured in various ways. For example, and with regards to a first option, a separate label may simply be used to identify a pseudowire group within an LSP. The association of an OAM message and a pseudowire group is straightforward. There could potentially be multiple pseudowire group labels per LSP. As a second option, one group label can be used to identify a common pseudowire group channel on the LSP. In this implementation, one pseudowire group label is provided per LSP. The OAM message association to a pseudowire group is not as simple as the first option. As a third option, one pseudowire is simply designated to convey grouping information (e.g., without using a group label). In this case, there is no need for a pseudowire group label. Again, the OAM message association to a pseudowire group is not as simple as the first option." Rustogi, ¶ [0023].<br><br>"Any combination of formatting (for the Group ID and the pseudowire group label) can be used in the group message to be communicated in the network. In one example, only one of these elements is communicated when an aggregate fault condition is detected, or these elements can be combined into a single unique identifier. In the most generic example, a group message would at least include the Group ID (identifying the pseudowires affected by the fault) and a pseudowire group label (identifying an aggregate channel for communicating the group message). In this generic sense, a pipe (the Group ID) within a pipe (the pseudowire group label) is being identified, where the group message is identifying both elements during an aggregate fault condition. Operational details of communication system 10 are described below with reference to FIGS. 2-6. Note that before turning to additional example flows and example embodiments of the present disclosure, a brief overview of the infrastructure of communication system 10 is provided." Rustogi, ¶ [0024].<br><br>"CE1 12, CE2 14, and CE3 16 represent devices, infrastructure, equipment, clients, or customers seeking to initiate a data session in communication system 10. These elements may can comprise a digital subscriber line access multiplexer (DSLAM), a router, a personal computer, a server, a switch, and/or other devices associated with data propagation. Further, these elements may sit behind, or in front of, one or more of these identified devices. The term 'CE' may be inclusive of the devices identified above (e.g., a DSLAM, a switch, etc.), as well as devices used to initiate a communication, such as a console, a proprietary endpoint, a |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | telephone, a cellular telephone, a bridge, a computer, a personal digital assistant (PDA), a laptop or an electronic notebook, or any other device, component, element, or object capable of initiating voice, audio, media, or data exchanges within communication system 10. The customer element may also include any device that seeks to initiate a communication on behalf of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating a voice, a video, text, or a data exchange within communication system 10. Data, as used herein in this document, refers to any type of video, numeric, voice, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another." Rustogi, ¶ [0025].<br><br>"SPE1 30, SPE2 32, TPE1 20, TPE2 22, and TPE3 24 are network elements that facilitate communications in two directions in a network environment. In one particular example, each of these network elements is a switch configured to exchange data over static and/or dynamic pseudowire links. Further, the traffic exchanged between these components may be directed over an MPLS transport in certain embodiments. As used herein in this Specification, the term 'network element' is meant to encompass switches, routers, bridges, gateways, servers, processors, loadbalancers, firewalls, or any other suitable device, component, element, or object operable to exchange or process information in a network environment. Moreover, these network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information. Along similar design alternatives, any of the internal modules and components of these network elements may be combined in various possible configurations." Rustogi, ¶ [0029].<br><br>"Turning to FIG. 2, FIG. 2 is a simplified block diagram of an example system 60 for providing an example use case using per-label switched path (LSP) pseudowire group labels. FIG. 2 includes a TPE1 62, a TPE2 64, a TPE3 66, a SPE1 68, and a SPE2 70. Each pseudowire group is identified, where a group identification (ID) for Group A and Group B is depicted at TPE1 62. Similarly, Groups C, D, and E have Group IDs at SPE1 68. TPE2 64 and TPE3 66 can couple to interfaces C and D, respectively." Rustogi, ¶ [0031]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "In this particular example, interfaces A and B have failed. Note that there is a multitude of attachment circuits (e.g., 1000 attachment circuits) that are being transported over these interfaces A and B, where the attachment circuits are being tunneled into a corresponding number of pseudowires. For example, there could be 500 attachment circuits on interface A (implicating 500 pseudowires) and 500 attachment circuits on interface B, where the fault condition for the pseudowires should be signaled. In other flawed systems, an architecture would individually signal this fault condition for each pseudowire (e.g., via signaling between TPE1 62 and SPE1 68). Instead of sending 500 messages, a single message can be sent, where a single label (and Group ID) can be used to identify the pseudowires. In this case, the Group ID A is used to signal the fault condition for 300 pseudowires and for 200 pseudowires (i.e., the top two links connecting TPE1 62 and SPE1 68) using a single message (that includes Group Label A and Group ID A). Thus, the status for Group A is quickly communicated to SPE1 68. Similarly, Group ID B can be used to signal the status of the other 500 pseudowires to appropriately convey the status for Group B. More specifically, the message can include Group Label A and Group ID B. Note that all 1000 pseudowires have effectively been accounted for using these Group IDs A and B." Rustogi, ¶ [0032].<br><br>"FIG. 3 is a simplified block diagram of an example system 72 for providing another use case for pseudowire group labels. Note that the grouping mechanism outlined herein is not limited to pseudowires that propagate over LSPs. Certain pseudowires can propagate over an LSP and represent one group, where two ports can be provisioned for two different groups (e.g., Group A and Group B). Hence, FIG. 3 is depicting a use case using pseudowire group labels for <port, LSP>mapping. In a general sense, such a configuration is showing how pseudowire mechanics can be used to offer different group signaling, which may be based on various possible implementations. Thus, there is a group level construct corresponding to the group labels that are created such that any OAM protocol can send the appropriate aggregate messages. In this particular example, the signaling for Group ID A, B, C, and D is similar to that of FIG. 2; however, the grouping mechanism has simply changed." Rustogi, ¶ [0035]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "FIG. 4 is a simplified block diagram of an example system 76 for providing another use case for pseudowire group labels. In this particular example, interface C fails (as shown at TPE2 64). Note that the same logical flow occurs in FIG. 4 in terms of the group signaling, as previously discussed. The group labels in two directions do not have to be the same, where the groupings for the messaging are not necessarily symmetrical. In this particular example, TPE2 64 sends a status for Group E with the corresponding group label (i.e., Group ID E for 300 pseudowires), where that message will have a Group Label E and a Group ID E. Hence, this particular signaling is indicative of 300 pseudowires failing in the network. SPE1 68 can send the status for Group F (where the Group ID F is associated with 300 pseudowires) to TPE1 62, where that message includes a Group Label F and a Group ID F." Rustogi, ¶ [0036].<br><br>"FIG. 5 is a simplified block diagram of an example system 80 for providing another use case for pseudowire group labels. In this particular example, interface D fails (as shown at TPE3 66), where all 700 pseudowires fail. In one implementation, TPE3 66 does not have a 700 pseudowire Group ID. Instead, the Group IDs can correspond to 200 and 500 pseudowires, when summed together account for the 700 pseudowires. In this particular example, TPE3 66 sends one message for Group I (representing 200 pseudowires) and another message for Group J (representing 500 pseudowires) to SPE1 68. In response, SPE1 68 sends a message for Group G (representing 200 pseudowires) and another message for Group H (representing 500 pseudowires). Again, the signaling being exchanged between these elements is minimal due to the effective grouping of pseudowires. SPE1 68 also sends a single message for Group I (associated with 200 pseudowires) and Group J (associated with 500 pseudowires) to TPE3 66, which is coupled to interface D. Group ID G is associated with 200 pseudowires, whereas Group ID H is representative of 500 pseudowires." Rustogi, ¶ [0037].<br><br>"FIG. 6 is a simplified table 74 illustrating an example set of pseudowire group provisioning parameters for TPE1 62, where these particular provisioning parameters could be relevant to the configuration of FIG. 3. At least in one generic sense, FIG. 2 can reflect one approach for mapping a PW group label to a PW Group ID, while FIGS. 3-5 can reflect a second approach for such mappings, where table 74 is associated with that second approach." Rustogi, ¶ [0038]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| | | "In particular, table 74 illustrates the mapping between SPE1 68 and TPE1 62. The first column represents the attachment circuit port (e.g., interface A, interface B, remote interface C on TPE2 64, and remote interface D on TPE3 66). Additionally, table 74 depicts a number of LSPs, a set of pseudowire grouping labels, and a set of pseudowire Group IDs. Note that the Group IDs are provided inside the pseudowire group labels in this example such that these two columns match in table 74. Additionally, note that table 74 is merely representing some of the possible characteristics in a single direction, where different constructs could be used in the reverse direction. Note that the provisioning as discussed herein can significantly reduce messaging such that these presented concepts offer increased scalability. This is due in part to the nominal processing that occurs in the network, in contrast to the processing required to evaluate a prolific amount of signaling messages associated with particular pseudowires. Additionally, the paradigm discussed herein can afford service providers an adequate amount of downtime after a failure has occurred in the network." Rustogi, ¶ [0039]. |
| 17[b] | determining an overall cost for each entity pair of said plurality of entities: | The Reference discloses determining an overall cost for each entity pair of said plurality of entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[b]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| 17[c] | selecting an entity pair from said plurality of transport entities based at least in part upon said overall cost; and | The Reference discloses selecting an entity pair from said plurality of transport entities based at least in part upon said overall cost.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[c]. |
| 17[d] | if an entity pair reselection event occurs, reselecting said entity pair from the group consisting of said entity pair and a replacement entity pair comprising at least one entity distinct from the entities of said entity pair, | The Reference discloses if an entity pair reselection event occurs, reselecting said entity pair from the group consisting of said entity pair and a replacement entity pair comprising at least one entity distinct from the entities of said entity pair.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[d]. |

| No. | '821 Patent Claim 17 | The Reference |
|---|---|---|
| 17[e] | wherein said entity pair reselection event is selected from a group consisting of adding an entity to said plurality of transport entities, removing an entity from said plurality of transport entities, an operational status change for one of said plurality of transport entities, and a change in overall cost for one of said plurality of transport entities. | The Reference discloses wherein said entity pair reselection event is selected from a group consisting of adding an entity to said plurality of transport entities, removing an entity from said plurality of transport entities, an operational status change for one of said plurality of transport entities, and a change in overall cost for one of said plurality of transport entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 1[e]. |

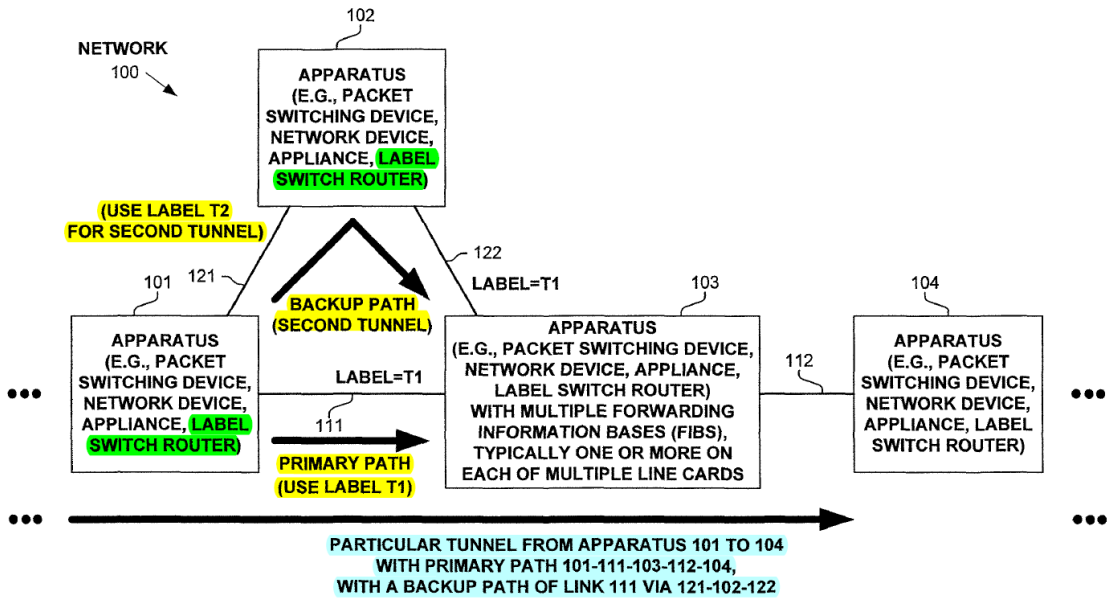
| No. | '821 Patent Claim 18 | The Reference |
|---|---|---|
| 18[preamble] | The non-transitory computer readable media of claim 17, wherein said step of selecting an entity pair further comprises: | The Reference discloses the non-transitory computer readable media of claim 17, wherein said step of selecting an entity pair further comprises.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

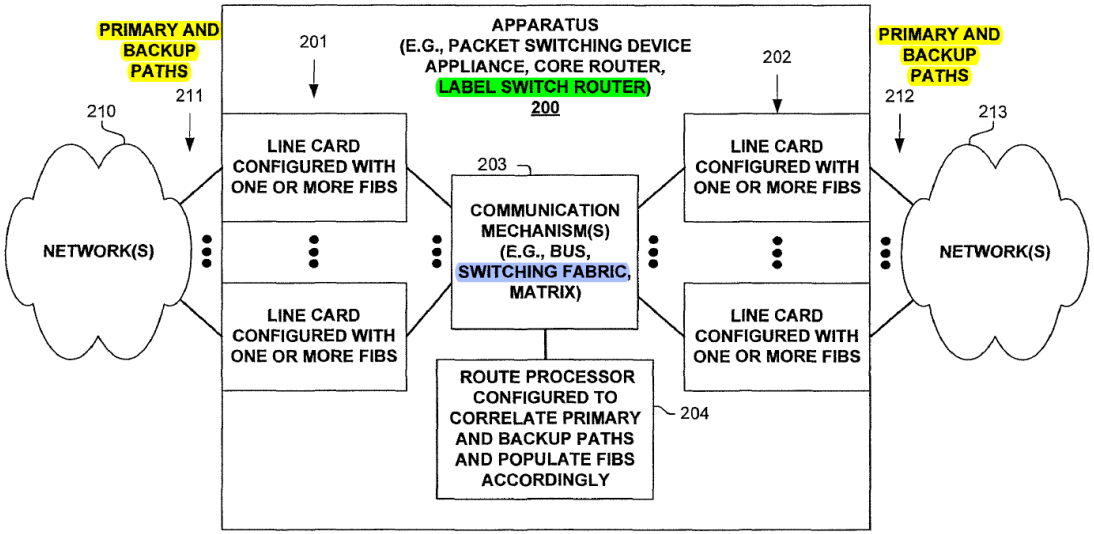
| No. | '821 Patent Claim 18 | The Reference |
|---|---|---|
| 18[a] | selecting a working entity from said plurality of transport entities; | The Reference discloses selecting a working entity from said plurality of transport entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 2[a]. |
| 18[b] | selecting a protection entity from said plurality of transport entities; and | The Reference discloses selecting a protection entity from said plurality of transport entities.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 2[b]. |
| 18[c] | selecting an active entity from the set consisting of said working entity and said protection entity. | The Reference discloses selecting an active entity from the set consisting of said working entity and said protection entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br>*See* Claim 3. |

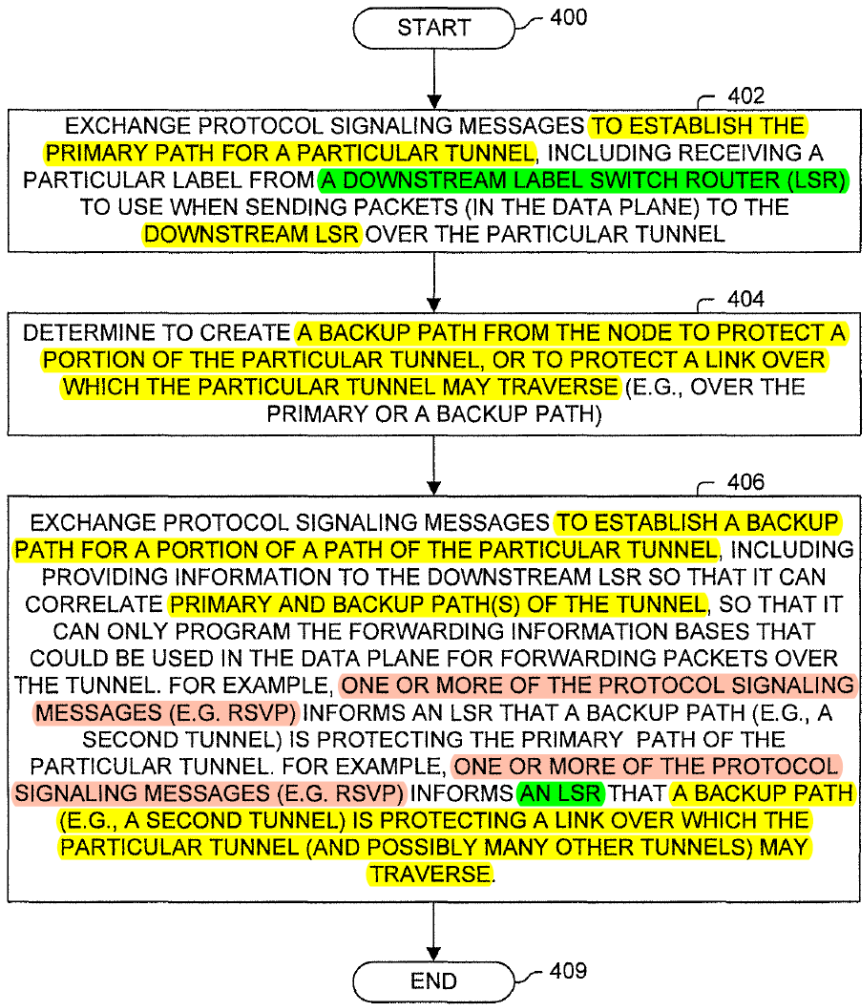
| No. | '821 Patent Claim 19 | The Reference |
|---|---|---|
| 19 | The non-transitory readable media of claim 18, wherein said step of selecting an entity pair further comprises minimizing an overall cost function. | The Reference discloses the non-transitory readable media of claim 18, wherein said step of selecting an entity pair further comprises minimizing an overall cost function.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 4. |

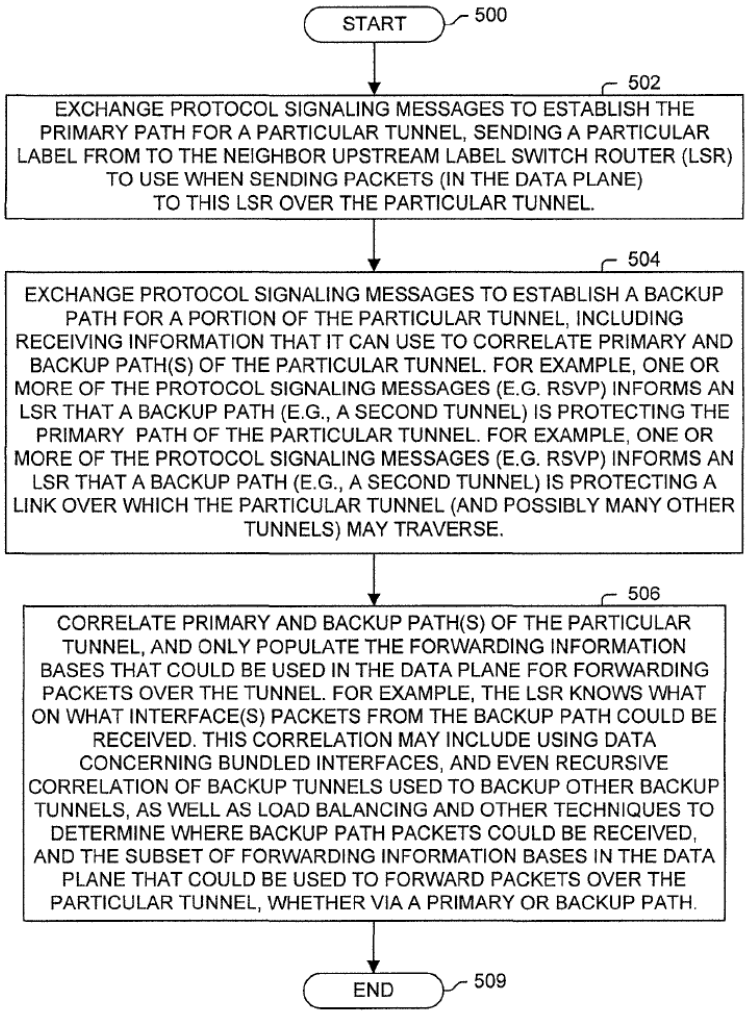
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| 20[preamble] | The non-transitory readable media of claim 19, wherein said overall cost function comprises: | The Reference discloses the non-transitory readable media of claim 19, wherein said overall cost function comprises.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |
| 20[a] | minimizing a probability of concurrent failure of said protection entity and said working entity; and | The Reference discloses minimizing a probability of concurrent failure of said protection entity and said working entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | *See* Claim 5. |
| 20[b] | a predefined metric selected from the group consisting of interior gateway protocol (IGP) and traffic engineering (TE). | The Reference discloses a predefined metric selected from the group consisting of interior gateway protocol (IGP) and traffic engineering (TE).<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Cisco IOS system, Juniper IOS System, IETF MPLS-TP System, Doshi '239, Sivabalan '928, and Zamfir '948.<br><br>*See* Claim 7.<br><br>Cisco created and developed the MPLS and MPLS-TE standards and patented technology based on those standards *before* Orckit utilized such technology. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Filsfils<br>• Taylor<br>• Vasseur '879<br>• Rustogi<br><br>**Filsfils discloses**:<br>"In one embodiment, forwarding information bases (FIBs) are selectively populated in a packet switch. A packet switching device determines, based on one or more protocol signaling messages, a subset, which is less than all, on which FIBs a lookup operation may be performed for identifying forwarding information for a received particular packet. The packet switching device populates each of these FIBs, but not all of the FIBs of the packet switching device, with forwarding information corresponding to the particular forwarding value. Thus, FIB resources are consumed for only those FIBs which could actually be used, and not all of the FIBs, for forwarding packets in the data plane of the packet switching device, whether these packets are received on a primary or backup path." Filsfils, Abstract. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 1**<br>Filsfils, FIG. 1 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIGURE 2<br>Filsfils, FIG. 2 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 4**<br>Filsfils, FIG. 4 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>**FIGURE 5**<br>Filsfils, FIG. 5. |

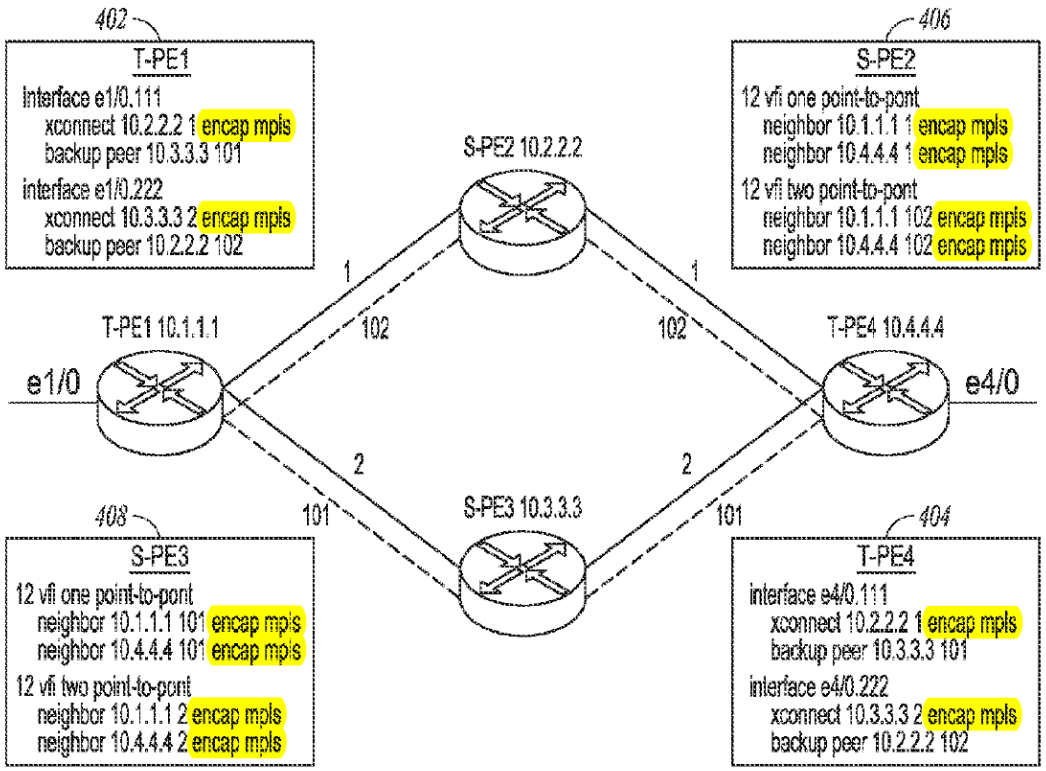
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "The communications industry is rapidly changing to adjust to emerging technologies and ever increasing customer demand. This customer demand for new applications and increased performance of existing applications is driving communications network and system providers to employ networks and systems having greater speed and capacity (e.g., greater bandwidth). In trying to achieve these goals, a common approach taken by many communications providers is to use packet switching technology, including switching packets on labels especially in the core network using Multiprotocol Label Switching (MPLS)." Filsfils, 1:12-22.<br><br>"Tunnels, such as MPLS-TE (Traffic Engineering) and MPLS-TP (Transport Profile), are paths established through a network in order to transport packets efficiently through a label switched network. Fast Re-Route (FRR) is a technology that allows backup paths to be established in the network, which can be used in case of a problem with a primary path (original primary path or currently used backup path) of the tunnel. RFC 4090, entitled "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," provides an extension of the protocol signaling to establish backup label switched path (LSP) tunnels for local repair of LSP tunnels." Filsfils, 1:23-33.<br><br>"Expressly turning to the figures, FIG. 1 illustrates a network 100 operating according to one embodiment. Shown are four apparatus 101-104 (e.g., packet switching devices such as a label switch router, network device, and/or appliance). For explanation purposes, each of apparatus 101-104 will be referenced as a label switch router (LSR)." Filsfils, 5:41-46.<br><br>"As shown, a particular tunnel is established, using a signaling protocol and exchanging of protocol signaling messages. Note, LSR 101 may, or may not, be an endpoint of the particular tunnel (e.g., LSR 101 may be an intermediate LSR on the path of the particular tunnel). The primary path of the particular tunnel includes spans from LSR 101 via link 111 to LSR 103 and via link 112 to LSR 104. Note, LSR 104 may be an intermediate LSR on the path of the particular tunnel, or an endpoint of the particular tunnel. Further, for this example embodiment, LSR 103 signals LSR 101 to use label T1 at the top of the label stack in the header of a packet sent to it on the particular tunnel." Filsfils, 5:47-58.<br>"A second tunnel from LSR 101 via link 121 to LSR 102 and via link 122 to LSR 103 is similarly configured using a signaling protocol and exchanging of protocol signaling |

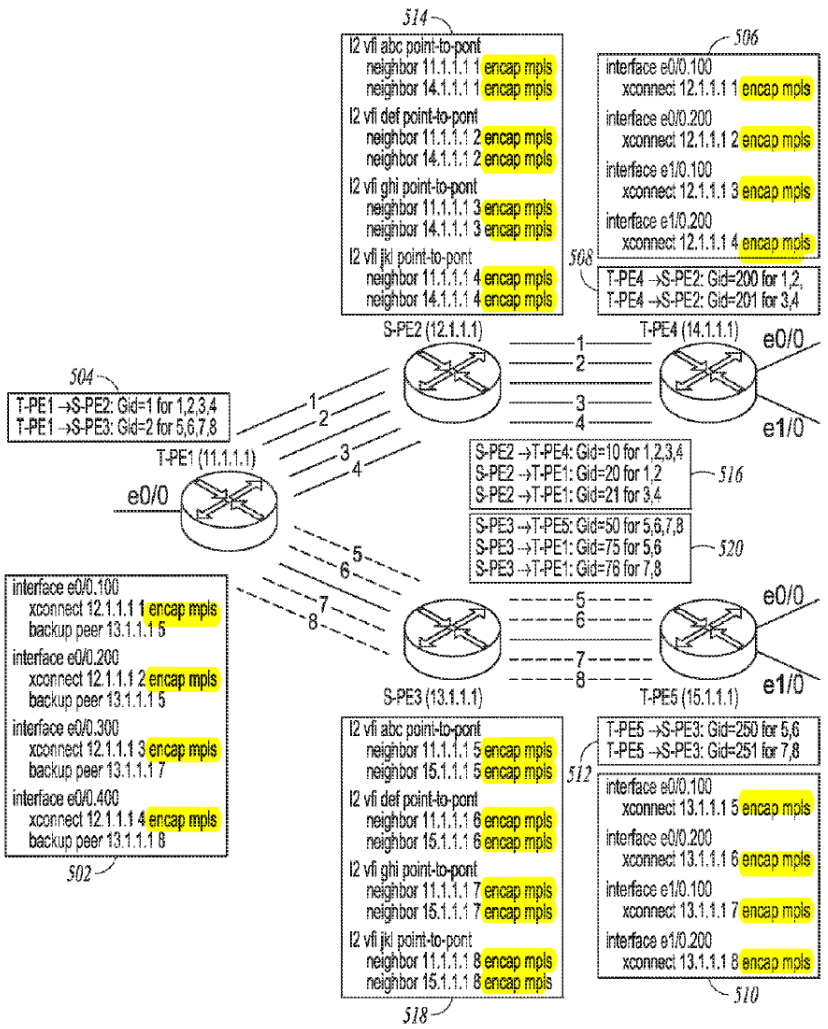
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | messages. For example purposes, LSR 102 signals LSR 101 to use label (T2) at the top of the label stack in the header of a packet sent to it on the second tunnel. In one embodiment, LSR 101 creates the second tunnel in response to determining, or being instructed to, create a backup path to protect link 111 and/or protect all or certain tunnels traversing link 111." Filsfils, 5:59-67.<br><br>"As shown in FIG. 1, link 111 (primary path of the particular tunnel and/or all or certain tunnels traversing link 111) is protected by LSR 101 using the second tunnel (backup path). When sending packets over the particular tunnel over link 111, LSR 101 includes label T1 at the top of the label stack of these packets. If link 111 cannot be used for communicating packets of the particular tunnel, LSR 101 sends packets over the backup path for the particular tunnel by sending packets to LSR 102, with these packets having a label stack including: label T2 followed by label T1. Thus, LSR 102 will receive these packets, pop the top label (T2) from the label stack of each of these packets, identify based on label T2 to send these packets to LSR 103. After popping the top label, the label at the top of the label stack of these packets is T1, which is the same label LSR 103 expects to receive for the particular tunnel. Therefore, these packets received with label T1 at the top of their label stack, are forwarded (after popping label T1 from their label stack) by LSR 103 over the particular tunnel to LSR 104." Filsfils, 6:6-24.<br><br>"One embodiment acquires such additional information by extending Resource Reservation Protocol (RSVP) to provide information which allows a packet switch to correlate primary and backup paths. Thus, a packet switch can use this additional information in determining which of its forwarding information bases (FIBs) could possibly be used in forwarding packets (e.g., in the data plane of the packet switch)." Filsfils, 6:51-57.<br><br>"In providing this additional information to LSR 103, one embodiment communicates an extended RSVP message (including a new or modified RSVP object) or other message to LSR 103 on the second tunnel. This messages designates one or more primary tunnels (e.g., label T1 in our example) and/or a link (e.g., link 111). As LSR 103 knows what interface that it received this message, LSR 103 knows that it must populate forwarding information for these primary tunnels, either specified (e.g., by a label such as T1), or all labels corresponding to |

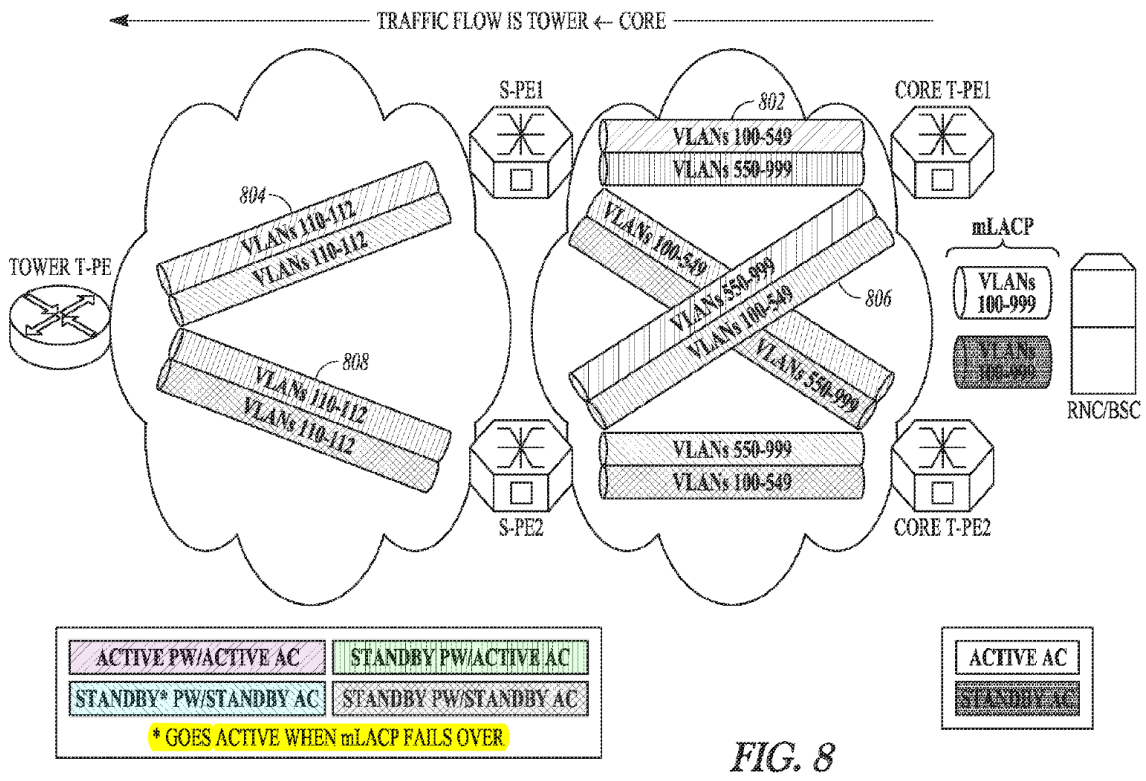
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | tunnels which could be received over link 111. In one embodiment, the extended RSVP or other message communicated to LSR 103 also includes an identification of the backup tunnel (e.g., T2) over which the RSVP or other message is being received, as the identification the tunnel over which a packet is received is often not communicated in a packet (e.g., in the case of Penultimate Hop Popping)." Filsfils, 7:19-34.<br><br>"As shown in FIG. 1, one embodiment includes apparatus 103, which populates less than all of its FIBs with forwarding information for a tunnel (although all FIBs may be populated for certain tunnels). One embodiment includes apparatus 101 and/or 102 which communicates, via a signaling protocol (e.g., an extension of RSVP, or using another protocol), information which allows apparatus 103 to determine the relationship between primary and backup paths, such that apparatus 104 can correlate this primary and backup path information (possibly also correlating backup path of backup path information, and/or bundled interfaces and/or bundled links) to identify a minimum subset of the FIBs that could possibly be used in forwarding packets of particular primary paths (e.g., tunnels)." Filsfils, 7:49-62<br><br>"Turning to FIG. 2, illustrates an apparatus 200 (e.g., packet switching devices such as a label switch router, network device, and/or appliance) operating in one embodiment. As shown, apparatus 200 includes line cards 201, 202 communicatively coupled via communication mechanism(s) 203 (e.g., bus, switching fabric, and/or matrix). Additionally, route processor 204 is configured to correlate primary and backup paths of tunnels, and to populate minimum subsets of FIBs with forwarding information for labels. Again, a minimum subset of FIBs for a particular path or label of the particular path is the set of FIBs that are determined to possibly be used in forwarding packets of a primary path, whether the label is received in a packet over the primary path or over a backup path, and possibly considering backup paths of a backup path and/or the possibly effect of bundled interfaces and/or bundled links." Filsfils, 7:63-8:11.<br><br>"As shown in FIG. 2, apparatus 200 is communicatively coupled via primary and backup paths 211, 212 to networks 210 and 213 (which could be the same network). As illustrated, each of line cards 201, 202 includes one or more FIBs. By correlating on which line card(s) 201, 202 and even within line cards 201, 202 that have multiple FIBs, primary and backup path(s) of tunnels, the number of FIB entries populated in apparatus 200 can typically be reduced, |

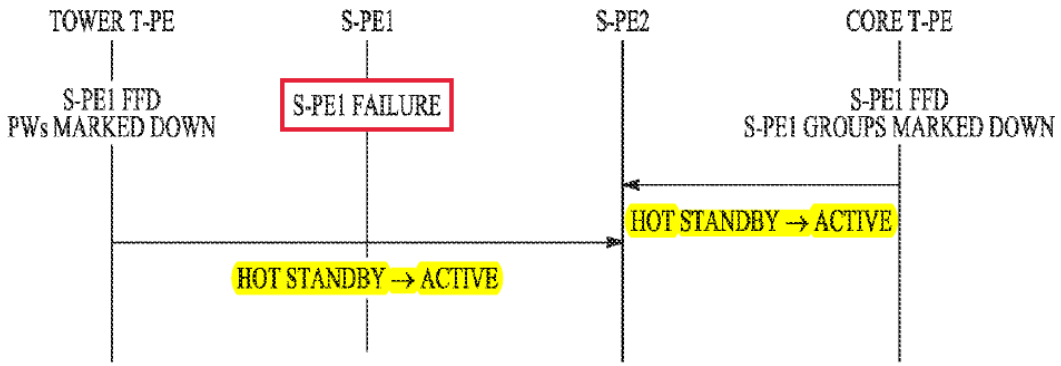
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | possibly significantly saving memory/storage resources and resources used to populate the FIBs." Filsfils, 8:12-21.<br><br>"FIG. 4 illustrates a process performed in one embodiment. Processing begins with process block 400. In process block 402, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes receiving a particular label for a downstream LSR to use when sending packets to the downstream LSR over the particular tunnel." Filsfils, 8:61-67.<br><br>"In process block 404, a determination is made to create a backup path from the node (e.g. the node performing these operations). This backup path may be used to protect one or more particular tunnels, and/or may be used to protect a link which is used to carry packet traffic of one or more tunnels." Filsfils, 9:1-5.<br><br>"In process block 406, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including providing information to the downstream LSR so that the downstream LSR can correlate primary and backup path(s) of the particular tunnel and substantially only program the FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs an LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:6-23.<br><br>"FIG. 5 illustrates a process performed in one embodiment. Processing begins with process block 500. In process block 502, protocol signaling messages are exchanged to establish the primary path for a particular tunnel, which typically includes sending a particular label for an upstream LSR to use when sending packets over the particular tunnel to this apparatus (e.g., an LSR performing these operations)." Filsfils, 9:26-32. |

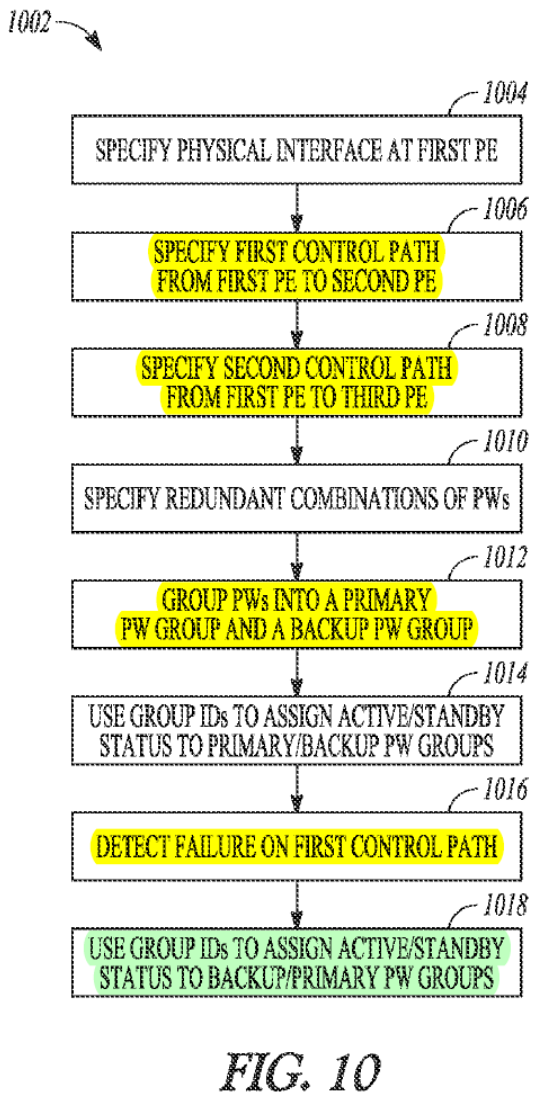
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "In process block 504, protocol signaling messages are exchanged to establish a backup path for a portion of the particular tunnel, including receiving information that the LSR can use to correlate primary and backup path(s) of the particular tunnel. For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting the primary path of the particular tunnel (and the LSR knows on which interface and/or link this protocol signaling message was received) For example, one or more of the protocol signaling messages (e.g. RSVP) informs the LSR that a backup path (e.g., a second tunnel) is protecting a link over which the particular tunnel (and possibly many other tunnels) may traverse (and the LSR knows on which interface and/or link this protocol signaling message was received)." Filsfils, 9:33-47.<br><br>"In process block 506, the primary and backup path(s) of the particular tunnel are correlated to identify the set of FIBs that could possibly be used in forwarding packets of the particular tunnel. Substantially only those FIBs that could potentially be used in the data plane for forwarding packets over the particular tunnel (either through a primary or backup path) are populated with the forwarding information (e.g., an entry corresponding to the label it advertised to use for the particular tunnel) for the particular tunnel. For example, the LSR knows what on what interface(s) packets from the backup path could be received. This correlation may include using data concerning bundled interfaces, and even recursive correlation of backup tunnels used to backup other backup tunnels, as well as load balancing and other techniques to determine where backup path packets could be received, and the subset of forwarding information bases in the data plane that could be used to forward packets over the tunnel, whether via a primary or backup path." Filsfils, 9:48-65.<br><br>**Taylor discloses**:<br>"Grouping pseudowires based on hardware interfaces and configured control paths enables improved pseudowire failover performance. Signaling status changes (e.g., from standby to active status) is facilitated by using group IDs for the pseudowire groups, thereby enabling improved failover performance when there is disruption in the network." Taylor, Abstract. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Taylor, FIG. 4 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 5*<br>Taylor, FIG. 5 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br>Taylor, FIG. 8 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 9*<br>Taylor, FIG. 9 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 10*<br>Taylor, FIG. 10 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "The present disclosure relates generally to communication networks and more particularly to pseudowire configurations in communication networks." Taylor, 1:8-10.<br><br>"Virtual Private Network (VPN) services provide secure network connections between different locations. A company, for example, can use a VPN to provide secure connections between geographically dispersed sites that need to access the corporate network so that each customer edge (CE) end point or node can communicate directly and independently with all other CE nodes. Different types of VPNs have been classified by the network layer used to establish the connection between the customer and provider network. For example, Virtual Private LAN Service (VPLS) is an architecture that delivers a multipoint Layer 2 VPN (L2VPN) service that in all respects emulates an Ethernet Local Area Network (LAN) across a wide metropolitan geographic area. All services in a VPLS appear to be on the same LAN, regardless of location. In other words, with VPLS, customers can communicate as if they were connected via a private Ethernet segment, i.e., multipoint Ethernet LAN services." Taylor, 1:12-28.<br><br>"In this context, each CE device at a customer site is connected to the service provider network at a provider edge PE) device by an Attachment Circuit (AC) that provides the customer connection to a service provider network, that is, the connection between a CE node and its associated PE node. Within the provider network, each PE device includes a Virtual Switch Instance (VSI) that emulates an Ethernet bridge (i.e., switch) function in terms of Media Access Control (MAC) address learning and forwarding in order to facilitate the provisioning of a multipoint L2VPN. A pseudowire (PW) is a virtual connection between two PE devices that connect two attachment circuits. In the context of the VPLS service, a pseudowire can be thought of as a point-to-point virtual link for each offered service between a pair of VSIs. Therefore, if each VSI can be thought of as a virtual Ethernet switch for a given customer service instance, then each pseudowire can be thought of as a virtual link connecting these virtual switches to each other over a Packet Switched Network (PSN) for that service instance." Taylor, 1:29-47. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "Since the failure of pseudowires obviously degrades network performance, some effort has been directed towards adding system redundancies including redundant pseudowires. However, the presence of redundant pseudowires alone is insufficient to improve overall failover performance, that is, the ability to switch over automatically to a redundant or backup system. Thus, there is a need for improved methods for managing pseudowires to facilitate pseudowire switching and enable improved failover performance." Taylor, 1:48-56.<br><br>"FIG. 4 shows details for PW connectivity in an exemplary network for an example embodiment." Taylor, 1:66-67.<br><br>"FIG. 5 shows details for PW grouping for an example embodiment." Taylor, 2:1-2.<br><br>"FIG. 8 shows an example network including redundant PW connections for an example embodiment." Taylor, 2:9-10.<br><br>"FIG. 9 shows an example sequence diagram for a failure mode related to the embodiment shown in FIG. 8." Taylor, 2:11-12.<br><br>"FIG. 10 shows a flowchart that illustrates a method of providing improved PW grouping according to an example embodiment." Taylor, 2:13-15.<br><br>"According to one embodiment, a method of providing improved pseudowire performance includes specifying a physical interface at a first PE node in a network, a first control path from the first PE node to a second PE node in the network, and a second control path from the first PE node to a third PE node in the network. With these specifications, the method then includes specifying redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node, and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Then these pseudowires can be grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. Group identifiers for the primary PW group and the backup PW group can then be used to assign an |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths. The method may then include detecting a failure on the first control path, and in response to the detected failure, using the group identifiers to assign the active status to the backup pseudowires and the standby status to the primary pseudowires." Taylor, 2:27-52.<br><br>"Pseudowires are used in pseudowire emulation edge-to-edge to provide a Layer 2 Virtual Private Network (L2VPN) connection. When large numbers (e.g., 4,000-6,000) of pseudowires are aggregated together on a single router, failure performance tends to be linear or O(n) where n is the number of pseudowires. While O(n) performance may be acceptable for small numbers of pseudowires, the effect on network outages can be increasingly undesirable as the number of pseudowires increases." Taylor, 2:54-62.<br><br>"For example, a cell-site router will typically start an approximately 2-minute procedure if contact with its controller, which is reached via a pseudowire, is lost for more than some threshold amount (e.g., between approximately 0.75 and 1.75 seconds in some cases). This can be a major impediment to the scalability of pseudowire deployments. These issues have become increasingly relevant as providers of Multiservice Broadband Networks (MBNs) are rapidly replacing or augmenting their traditional Synchronous Optical Networking (SONET) equipment with cheaper Ethernet equipment in the evolution towards a 4G (i.e., $4^{th}$ generation) network." Taylor, 2:63-3:7.<br><br>"One aspect of a solution to the problem of pseudowire failure is the deployment of redundant pseudowires. For example, redundant pseudowires have been used in the context of Multiprotocol Label Switching (MPLS) networks, which use a Label Distribution Protocol (LDP) to manage labels for forwarding traffic between routers. In this context, general requirements for redundancy schemes have been developed so that duplicate pseudowires are available when a given pseudowire fails (e.g., by using active/standby status indicators). In addition, more specific implementations for redundant pseudowires have also been developed." Taylor, 3:8-18. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 1 shows a reference network model 102 with applications to example embodiments disclosed herein. The reference network model 102 includes an aggregation network 104 of PE nodes and a distribution network 106 of PE nodes between a radio network controller (RNC) (or base station controller (BSC)) 108 on the core side of the model 102 and a radio tower 111 on the tower side of the model 102. Switching provider edge nodes S-PE1 and S-PE2 connect the two networks 104, 106. On the core side, two core terminating provider edges T-PE1 and T-PE2 connect to the RNC/BSC 108 through attachment circuits 110, 112. On the tower side, one tower terminating provider edge T-PE connects to the radio tower 111 through an attachment circuit 114." Taylor, 3:19-31.<br><br>"Additionally as noted in FIG. 1, peer-PE monitoring is carried out within each network 104, 106. That is, there is peer-PE monitoring between provider edges that share a segment, for example, by multi-hop bidirectional forwarding detection (BFD). Alternatively, peer monitoring can be accomplished by other means (e.g., MPLS-TP (Transport Protocol) keep-olives). This peer-PE monitoring is used to provide the mechanism for fast failure detection. Once a failure is detected, the network can react by "rerouting" the failed pseudowires to pre-provisioned backup paths and thus provide a minimal disruption in service to the end-user. This rerouting can be accomplished by LDP signaling between provider edges." Taylor, 3:32-44.<br><br>"The reference network model 102 may be considered as part of a larger hub-and-spoke model as shown in FIG. 2. A hub-and-spoke distribution model 202 includes a core network 204, distribution networks 206, and aggregation networks 208. Network elements including distribution nodes, aggregation nodes, and towers are also shown with nominal count values (e.g., 30 distribution nodes between the core network 204 and a distribution network 206). In this model 202, tower T-PEs are the spokes white core-PEs constitute the hub. Dozens to hundreds of tower T-PEs connect to a few S-PEs; these S-PEs are quite similar to ASBRs as they act as forwarders between the two distinct MPLS domains, providing isolation and, in the case of mobility, aggregation services. Typically, several aggregation networks 208 are connected to a single distribution network 206, eventually connecting the tower with the core router that connects the tower's ACs to the RNC/BSC. There are typically several distribution |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | networks in a Radio Access Network (RAN) connected to the service provider's core Internet Protocol (IP) network." Taylor, 4:11-30.<br><br>"With reference to FIG. 1, FIG. 3 shows a variety of failure modes encountered in the reference network model 102. Failure 302 of communications between tower T-PE and the S-PE can be detected via peer monitoring when both the S-PE and the T-PE are still active/alive. For example, this failure may be due to a toss of connectivity when the BFD session goes down. Failure 304 of S-PE1 can be due to a hardware failure, power outage, or the lack of BED-session maintenance capability (e.g., if the BFD-session hello timers cannot be serviced for the prescribed period of time). Failure 306 of communications between S-PE1 and core T-PE1 can be detected via peer monitoring when both S-PE1 and core T-PE1 are still active/alive. Failure 308 at core T-PE1 can be due to a hardware failure, power outage, or the lack of BFD-session maintenance capability." Taylor, 4:46-60.<br><br>"Pseudowire connectivity is further illustrated in FIG. 4 where the illustrated network includes four nodes: T-PE1 (10.1.1.1), S-PE2 (10.2.2.2), S-PE3 (10.3.3.3), and T-PEA (10.4.4.4). For the terminating nodes T-PE1 and T-PE4, specifications for VLANs (virtual Local Area Networks) connections (i.e., pseudowires) are shown using the Internet Operating System Command Line Interface (IOS CLI). The specification 402 for T-PE1 defines two VLANS as primary/backup combinations of virtual circuits for the network. The first three lines of the specification 402 define "VLAN 111" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 111" in the first line. The second line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE2 (10.12.2) with a virtual circuit Identification (VCID) set as VCID=1, and the third line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=101. The next three lines of the specification 402 define "VLAN 222" beginning with a specification of the hardware interface e1/0 of T-PE1 (10.1.1.1) and the label for "VLAN 222" in the fourth line. The fifth line uses the "xconnect" statement to specify a virtual circuit from that interface to S-PE3 (10.3.3.3) with VCID=2, and the sixth line uses the "backup peer" statement to specify another virtual circuit from that interface to S-PE2 (10.2.2.2) with VCID=102. These virtual circuits, VCID=1, VCID=2, VCID=101 and VCID=102 are shown in the figure between T-PE1 and the S-PEs with a solid line for the primary circuits VCID=1 |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | and VCID=2 and a dashed line for the backup circuits VCID=101 and VCID=102." Taylor, 4:61-5:23.<br><br>"With respect to T-PE1 in FIG. 4, although "VLAN 111" and "VLAN 222" share the same hardware port, they do not share the same "control path disposition." That is, "VLAN 111" is primary to S-PE2 (VCID=1) and standby to S-PE3 (VCID=101), white "VLAN 222" has an opposite configuration since it is primary to S-PE3 (VCID=2) and standby to S-PE2 (VCID=102). As discussed below, certain embodiments group pseudowires according to "control path disposition" (e.g., xconnect configuration as well as the hardware interface in order to improve failover performance. That is, to deal with both hardware port failures and switching path failures, the grouping criteria also considers the cross connects. In this case, on T-PE1 as well as T-PE4, there would exist two groups: one for active to S-PE2 and standby to S-PE3 and another for active to S-PE3 and standby to S-PE2 (i.e., the inverse configuration)." Taylor, 5:61-6:9.<br><br>"First, local connectivity is characterized by local group identifications (Group-IDs), which depend on whether the allocation is done at a T-PE or S-PE. FIG. 5 shows an embodiment that illustrates an allocation of local group IDs in a network including terminating nodes T-PE1 (11.1.1.1), T-PE4 (14.1.1.1), and T-PE5 (15.1.1.1) and switching nodes S-PE2 (12.1.1.1) and S-PE3 (13.1.1.1). The specification 502 for T-PE1 determines corresponding local group IDs 504 based on the hardware interface and the control path. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, and the Group-ID=2 corresponds to VCID=5, VCID=6, VCID=7, and VCID=8." Taylor, 6:10-21.<br><br>"Local group IDs are maintained in a database so that pseudowire redundancy is also maintained. First, in a case without pseudowire redundancy, all the xconnect configurations from the same physical interface to the same peer are assigned the same local group ID. So, for example, in Ethernet cases all xconnect configurations under sub-interfaces of the same physical interface to the same peer will be assigned the same local group ID (e.g., e0/0 and e0/1 are sub-interfaces of e0). FIG. 6 shows a database representation for T-PE4 from FIG. 5. From the root node 602 for T-PE4, there is a first interface node 604 for e0/1 and a second interface node 606 for e1/0. The first interface node 604 is configured towards a single peer |

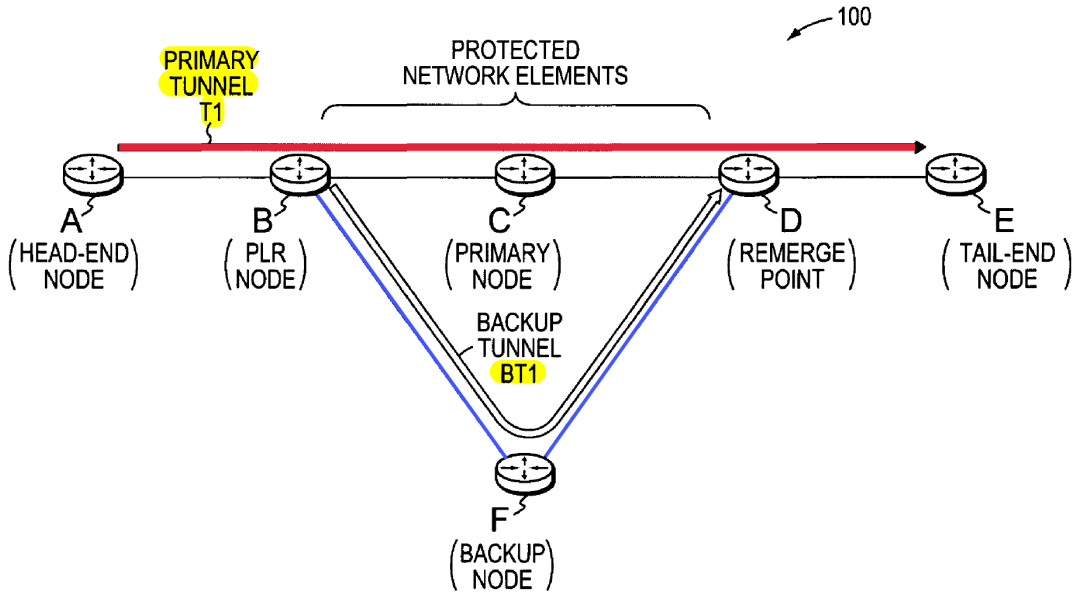
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | node (12.1.1.1) 608 and is thus assigned a single local group ID (Group-ID=200) 610. Similarly, the second interface node 606 is configured towards a single peer node (12.1.1.1) 612 and is thus assigned a single local group ID (Group-ID=201) 614. In this case from the assignment of local group IDs 508 in FIG. 5, Group-ID=200 corresponds to VCID=1 and VCID=2, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). Both pseudowires (VCID=1 and VCID=2) are assigned the same local group ID (Group-ID=200) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly, from the assignment of local group IDs 508 in FIG. 5, T-PE4 has VCID=3 and VCID=4 under the physical interface e1/0 going to the same peer S-PE2 (12.1.1.1), and the local Group ID (Group-ID=201) is assigned to these VCs." Taylor, 6:48-7:8.<br><br>"For the pseudowire redundancy case, a separate redundancy-group database is maintained by the xconnect application. This redundancy-group database contains the peer IDs in the group and the local group IDs advertised to them. This is needed to maintain a 1:1 mapping between the primary pseudowires and their corresponding backup pseudowires. FIG. 7 shows a database representation for T-PE1 from FIG. 5. From the root node 702 for T-PE1, there is an interface node 704 for e0/0 and a redundancy group node 706 that shows connections for configurations to a first peer node (12.1.1.1) 708, which is assigned a local group ID (Group-ID=1) 710, and a second peer node (13.1.1.1) 712, which is assigned a local group ID (Group-ID=2) 714. In this case, Group-ID=1 corresponds to VCID=1, VCID=2, VCID=3, and VCID=4, which are under hardware interface e0/0 and configured towards the same peer, S-PE2 (12.1.1.1). These pseudowires are assigned the same group ID (Group-ID=1) in this case, and this is advertised in label mapping messages towards the remote provider edge, i.e., S-PE2. Similarly from the assignment of local group IDs 504 in FIG. 5, T-PE1 has VCID=5, VCID=6, VCID=7, and VCID=8 under the physical interface e0/0 going to another peer S-PE3 (13.1.1.1), and the local group ID (Group-ID=2) is assigned to these VCs. In this case these local group IDs are organized as a redundancy group 706." Taylor, 7:9-35. |

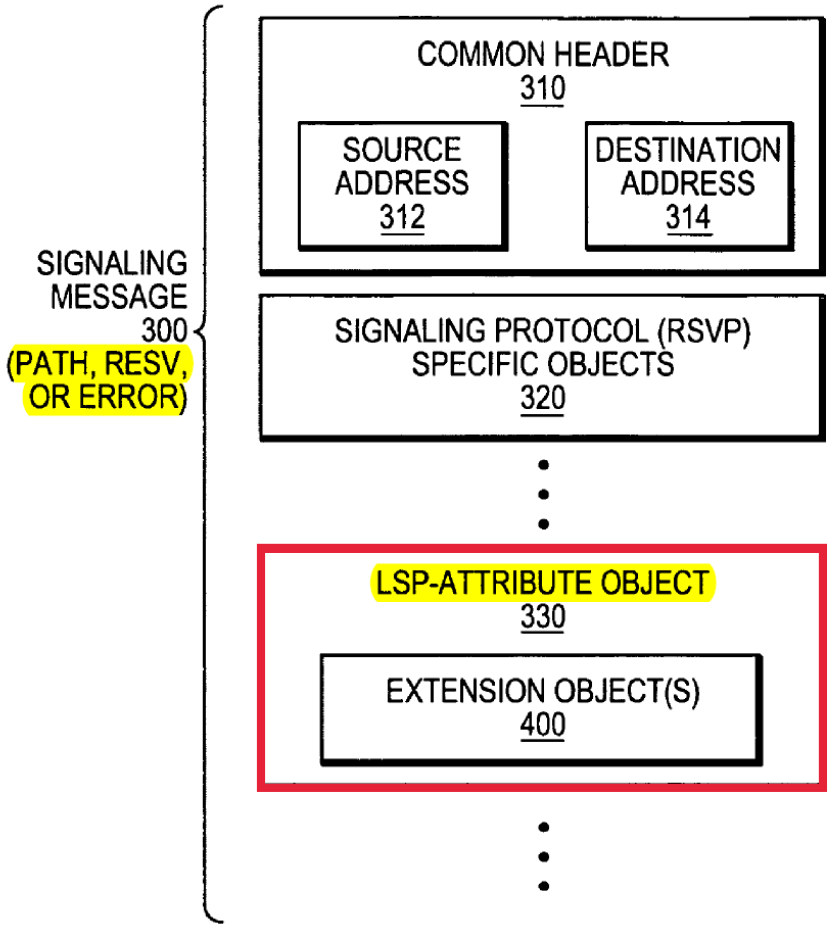
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "Pseudowire grouping allows multiple pseudowires to be aggregated together when signaling either label withdrawals or status changes between segment end-point provider edges. This signaling can be carried out through LDP grouping TLV (Type Length Value). For example, when a PE node fails (e.g., failure 304 of S-PE1), aggregating the inter-segment PE signaling using the grouping TLV can provide significant scaling advantages. This allows all pseudowires sharing a physical port and PW configuration (e.g., xconnect configuration) to be signaled en masse between segment-adjacent provider edges." Taylor, 7:36-46.<br><br>"FIG. 8 shows an example based on FIG. 1 where VLAN ACs are shown as grouped by both port/HW-interface and pseudowire-class. The grouping criterion allows all "similar" pseudowires to be signaled together: All the grouped pseudowires share the same port and next-hop provider edge. Additionally, the figure contains many pseudowires, each grouped into a shaded tube. For example, the tube labeled "VLANs 100-549" contains 450 pseudowires grouped together. This figure depicts an incoming Ethernet comprised of 900 VLANs being segmented in two with 450 VLANs (100-549) active to S-PE1 while the other half of the VLANs (550-999) being active to S-PE2. This might be considered a type of manual load balancing. Furthermore, the aggregation network is only showing a single tower and the VLANs associated with it; other VLAN destinations are not shown in the figure." Taylor, 7:47-62.<br><br>"'VLANs 110-112' are active along a first pseudowire path 802 from Core T-PE1 to S-PE1 and a second pseudowire path 804 from S-PE1 to Tower T-PE. When a failure occurs at S-PE1 (e.g., as the switching node failure 304 shown in FIG. 3), then the standby pseudowires become active for 'VLANs 110-112' along a first pseudowire path 806 from Core T-PE1 to S-PE2 and a second pseudowire path 808 from S-PE2 to Tower T-PE." Taylor, 7:63-8:3.<br><br>"The standby pseudowires in FIG. 8 can be configured as HSPWs, a configuration that enables ACs to quickly failover to pre-provisioned pseudowires that are in active state but set to not-forwarding. Then when a failure occurs, switching over to these pre-provisioned HSPWs occurs quickly by switching from not-forwarding status to forwarding status." Taylor, 8:4-9. |

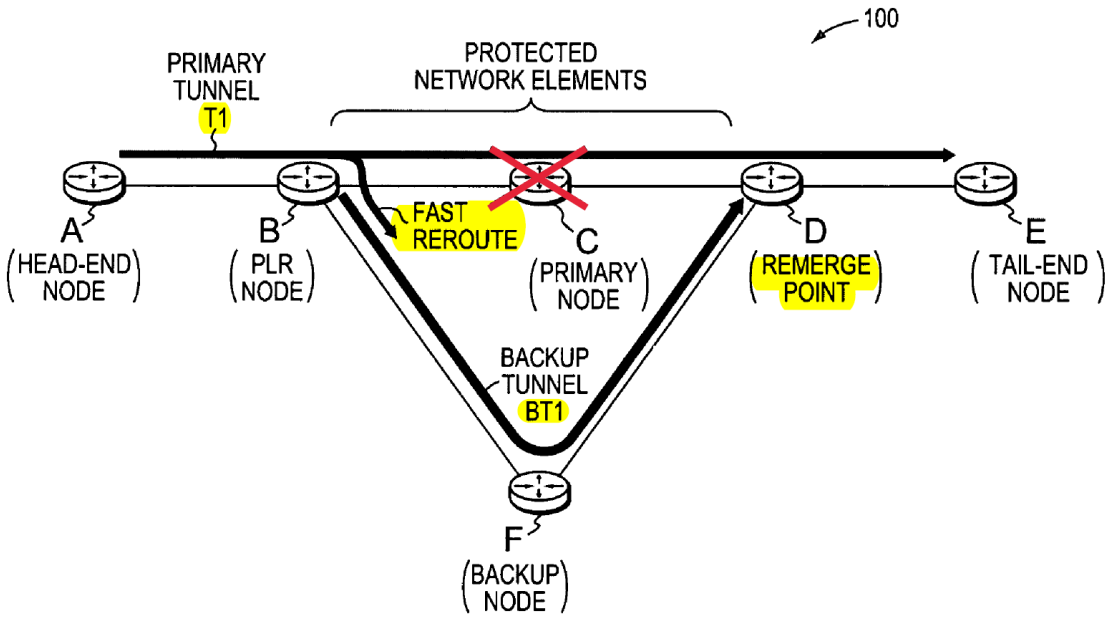
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 9 shows a UML (Unified Modeling Language) sequence diagram of signaling events related to the failover procedure illustrated in FIG. 8 for a switching node failure 304. The Tower T-PE and the Core T-PE independently detect a failure at S-PE1 (e.g., BFD forwarding with LDP signaling), and then mark down the status of the currently active pseudowires routed through S-PE1 and mark up the status of the standby pseudowires routed through S-PW2. Other failure modes shown in FIG. 3 can be handled similarly." Taylor, 8:10-18.<br><br>"In general, it is desirable for MPLS-based. Ethernet networks to react quickly to failures, so proactive detection mechanisms are employed in order to pick up system failures quickly. All proactive monitoring is typically done between PE peers on a single MPLS network. These provider edges on the edges of the MPLS network act similarly to an Autonomous System Boundary Router (ASBR). As a result, related embodiments detect control path failures, which may not be the same as pseudowire data path failures. That is, the data packets and control packets may take different paths between provider edges in a MPLS network although typically these paths are coincident. Thus, when the control and data paths are not coincident, if the control path fails, then all pseudowires utilizing the control path are marked as failed. As a corollary, if the data path fails and the control path remains healthy, then failure will not be detected from monitoring the control path." Taylor, 8:19-35.<br><br>"A failure of a monitored provider edge initiates a switchover of all active pseudowires using the failing provider edge to their configured HSPWs (if they exist). Grouping can greatly reduce the number of messages needed between provider edges (Inter-PE Aggregation) and within a single provider edge (Intra-PE Aggregation). Furthermore, the MPLS network itself may be internally resilient deploying technologies such as, but not limited to, MPLS-TE (MPLS Traffic Engineering) and ERR (Fast Reroute). The paths across the MPLS network may recover quickly and might not trip the fault-monitoring systems." Taylor, 8:36-46.<br><br>"With reference to the above discussion, FIG. 10 shows a method 1002 of providing improved PW grouping according to an example embodiment. In a first operation 1004 of the method 1002, a physical interface is specified at a first PE node in a network. In a second operation 1006, a first control path is specified from the first PE node to a second PE node in the network. In a third operation 1008, a second control path is specified from the first PE node to a third |

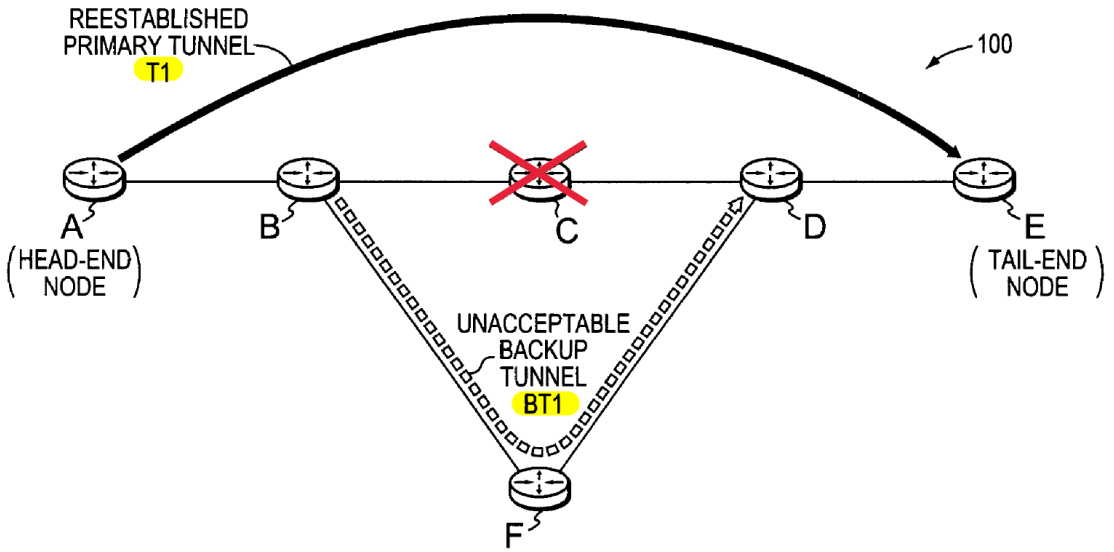
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | PE node in the network. These control paths related to a common physical interface can be used to characterize redundant pairs of pseudowires." Taylor, 8:48-58.<br><br>"In a fourth operation 1010, redundant combinations of pseudowires are specified, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node. Although a redundant combination may relate a single backup pseudowire to a given primary pseudowire, in some cases multiple backup pseudowires will be related to a given primary pseudowire for increased redundancy. In a fifth operation 1012, these pseudowires are grouped into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. In a sixth operation 1014, group identifiers for the primary PW group and the backup PW group are used to assign an active status to the primary pseudowires and a standby status to the backup pseudowires, where the active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 8:59-9:11.<br><br>"In an optional seventh operation 1016, a failure may be detected on the first control path, and in an optional eighth operation 1018, in response to the detected failure, the group identifiers may be used to assign the active status to the backup pseudowires and the standby status to the primary pseudowires. For example, the failure on the first control path may be detected by using BED packet streams between PE nodes of the network. Then the detected failure can be signaled to PE nodes in the network by sending LDP status updates between PE nodes in the network. Then, after receiving the failure detection signals, the group identifiers can be used again to assign the active status to the backup pseudowires and the standby status to the primary pseudowires by sending LDP status updates between PE nodes in the network." Taylor, 9:12-26.<br><br>"Typically the network in is an MPLS network and the PE nodes are routers that provide network services to connected CE nodes of a customer network. In general, each control path is an Internet Protocol (IP) routing path between PE nodes in the network and each data path is a label switched path (LSP) between PE nodes in the network." Taylor, 9:27-32. |

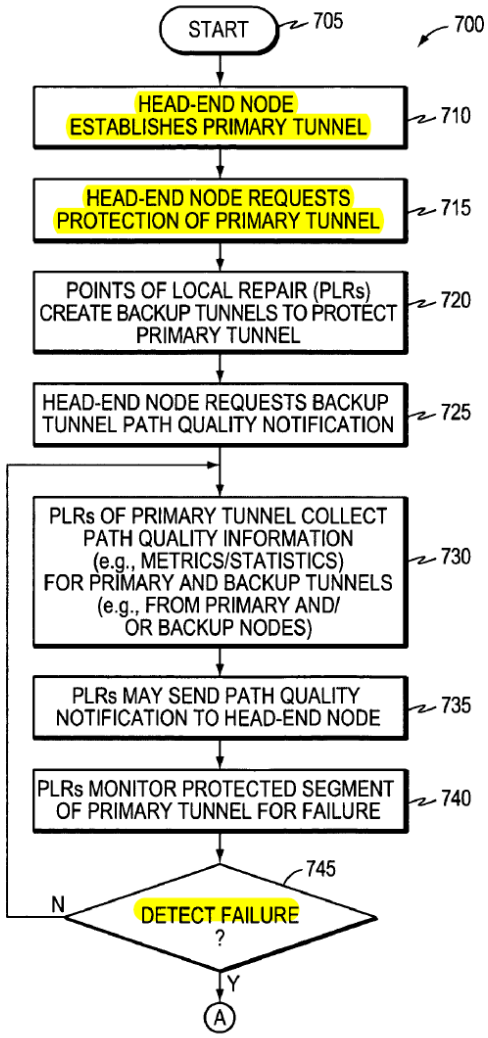
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 11 shows a schematic representation of an apparatus 1102, in accordance with an example embodiment. For example, the apparatus 1102 may be used to implement the method 1002 of providing improved pseudowire grouping as described above with reference to FIG. 10. The apparatus 1102 is shown to include a processing system 1104 that may be implemented on a server, client, or other processing device that includes an operating system 1106 for executing software instructions." Taylor, 10:2-10.<br><br>"In accordance with an example embodiment, the apparatus 1102 includes a PW management module 1108 that includes a first specification module 1110, a second specification module 1112, third specification module 1114, a fourth specification module 1116, a grouping module 1118, and an assignment module 1120. The first specification module 1110 operates to specify a physical interface at a first PE node in a network. The second specification module 1112 operates to specify a first control path from the first PE node to a second PE node in the network. The third specification module 1114 operates to specify a second control path from the first PE node to a third PE node in the network. The fourth specification module 1116 operates to specify redundant combinations of pseudowires, where each redundant combination includes a primary pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the second PE node and a backup pseudowire that is configured as a virtual circuit between the physical interface of the first PE node and the third PE node." Taylor, 10:11-29.<br><br>"The grouping module 1118 operates to group the pseudowires into a primary PW group that includes the primary pseudowires and a backup PW group that includes the backup pseudowires. The assignment module 1120 operates to use group identifiers for the PW groups to assign an active status to the primary pseudowires and a standby status to the backup pseudowires. The active status enables data transfers along corresponding PW data paths and the standby status disables data transfers along corresponding PW data paths." Taylor, 10:30-38. |

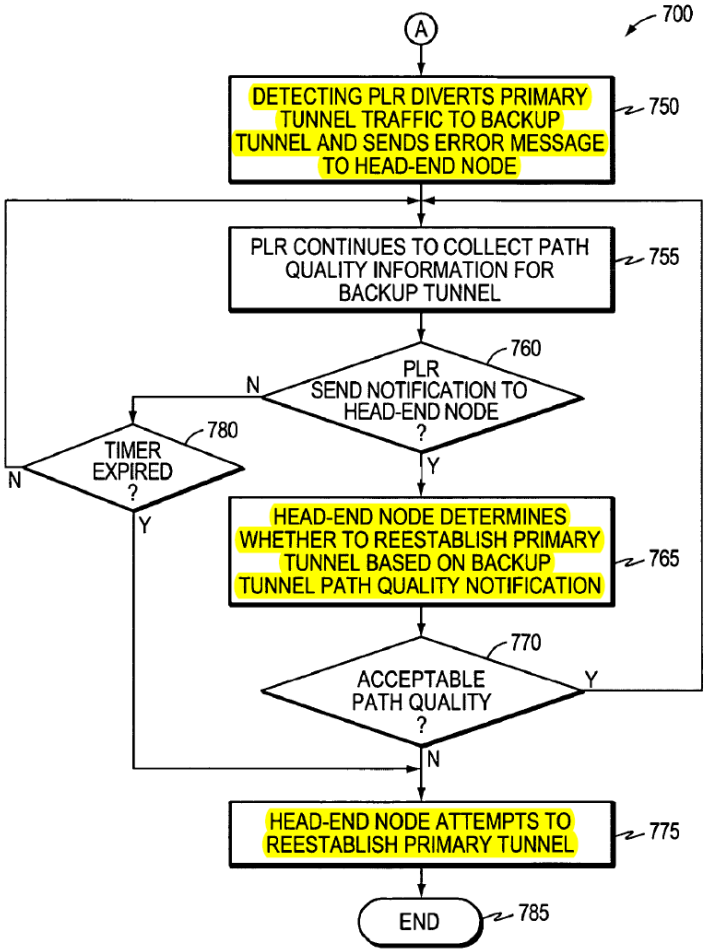
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | **Vasseur '879 discloses**:<br>"A technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, Abstract.<br>  <br><br>FIG. 1<br>Vasseur '879, FIG. 1 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Vasseur '879, FIG. 3 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Vasseur '879, FIG. 5 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 6<br>Vasseur '879, FIG. 6 (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 7A<br>Vasseur '879, FIG. 7A (annotated). |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 7B<br>Vasseur '879, FIG. 7B. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "The present invention relates to computer networks and more particularly to dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 1:10-14.<br><br>"Since management of interconnected computer networks can prove burdensome, smaller groups of computer networks may be maintained as routing domains or autonomous systems. The networks within an autonomous system (AS) are typically coupled together by conventional "intradomain" routers configured to execute intradomain routing protocols, and are generally subject to a common authority. To improve routing scalability, a service provider (e.g., an ISP) may divide an AS into multiple "areas." It may be desirable, however, to increase the number of nodes capable of exchanging data; in this case, interdomain routers executing interdomain routing protocols are used to interconnect nodes of the various ASes. Moreover, it may be desirable to interconnect various ASes that operate under different administrative domains. As used herein, an AS or an area is generally referred to as a "domain," and a router that interconnects different domains together is generally referred to as a 'border router.'" Vasseur '879, 1:40-56.<br><br>"An example of an interdomain routing protocol is the Border Gateway Protocol version 4 (BGP), which performs routing between domains (ASes) by exchanging routing and reachability information among neighboring interdomain routers of the systems. An adjacency is a relationship formed between selected neighboring (peer) routers for the purpose of exchanging routing information messages and abstracting the network topology. The routing information exchanged by BGP peer routers typically includes destination address prefixes, i.e., the portions of destination addresses used by the routing protocol to render routing ("next hop") decisions. Examples of such destination addresses include IP version 4 (IPv4) and version 6 (IPv6) addresses. BGP generally operates over a reliable transport protocol, such as TCP, to establish a TCP connection/session. The BGP protocol is well known and generally described in Request for Comments (RFC) 1771, entitled A Border Gateway Protocol 4 (BGP-4), published March 1995." Vasseur '879, 1:57-2:7. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "Examples of an intradomain routing protocol, or an interior gateway protocol (IGP), are the Open Shortest Path First (OSPF) routing protocol and the Intermediate-System-to-Intermediate-System (IS-IS) routing protocol. The OSPF and IS-IS protocols are based on link-state technology and, therefore, are commonly referred to as link-state routing protocols. Link-state protocols define the manner with which routing information and network-topology information are exchanged and processed in a domain. This information is generally directed to an intradomain router's local state (e.g., the router's usable interfaces and reachable neighbors or adjacencies). The OSPF protocol is described in RFC 2328, entitled OSPF Version 2, dated April 1998 and the IS-IS protocol used in the context of IP is described in RFC 1195, entitled Use of OSI IS-IS for routing in TCP/IP and Dual Environments, dated December 1990, both of which are hereby incorporated by reference." Vasseur '879, 2:8-24.<br><br>"An intermediate network node often stores its routing information in a routing table maintained and managed by a routing information base (RIB). The routing table is a searchable data structure in which network addresses are mapped to their associated routing information. However, those skilled in the art will understand that the routing table need not be organized as a table, and alternatively may be another type of searchable data structure. Although the intermediate network node's routing table may be configured with a predetermined set of routing information, the node also may dynamically acquire ("learn") network routing information as it sends and receives data packets. When a packet is received at the intermediate network node, the packet's destination address (e.g., stored in a header of the packet) may be used to identify a routing table entry containing routing information associated with the received packet. Among other things, the packet's routing information indicates the packet's next-hop address." Vasseur '879, 2:25-41.<br><br>"Multi-Protocol Label Switching (MPLS) Traffic Engineering has been developed to meet data networking requirements such as guaranteed available bandwidth or fast restoration. MPLS Traffic Engineering exploits modem label switching techniques to build guaranteed bandwidth end-to-end tunnels through an IP/MPLS network of label switched routers (LSRs). These tunnels are a type of label switched path (LSP) and thus are generally referred to as MPLS Traffic Engineering (TE) LSPs. Examples of MPLS TE can be found in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels dated December 2001, RFC 3784 |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | entitled Intermediate-System-to-Intermediate-System (IS-IS) Extensions for Traffic Engineering (TE) dated June 2004, and RFC 3630, entitled Traffic Engineering (TE) Extensions to OSPF Version 2 dated September 2003, the contents of all of which are hereby incorporated by reference in their entirety." Vasseur '879, 2:58-3:6.<br><br>"Establishment of an MPLS TE-LSP from a head-end LSR to a tail-end LSR involves computation of a path through a network of LSRs. Optimally, the computed path is the "shortest" path, as measured in some metric, that satisfies all relevant LSP Traffic Engineering constraints such as e.g., required bandwidth, "affinities" (administrative constraints to avoid or include certain links), etc. Path computation can either be performed by the head-end LSR or by some other entity operating as a path computation element (PCE) not co-located on the head-end LSR. The head-end LSR (or a PCE) exploits its knowledge of network topology and resources available on each link to perform the path computation according to the LSP Traffic Engineering constraints. Various path computation methodologies are available including CSPF (constrained shortest path first). MPLS TE-LSPs can be configured within a single domain, e.g., area, level, or AS, or may also span multiple domains, e.g., areas, levels, or ASes." Vasseur '879, 3:7-24.<br><br>"The PCE is an entity having the capability to compute paths between any nodes of which the PCE is aware in an AS or area. PCEs are especially useful in that they are more cognizant of network traffic and path selection within their AS or area, and thus may be used for more optimal path computation. A head-end LSR may further operate as a path computation client (PCC) configured to send a path computation request to the PCE, and receive a response with the computed path, potentially taking into consideration other path computation requests from other PCCs. It is important to note that when one PCE sends a request to another PCE, it acts as a PCC." Vasseur '879, 3:25-36.<br><br>"Some applications may incorporate unidirectional data flows configured to transfer time-sensitive traffic from a source (sender) in a computer network to a destination (receiver) in the network in accordance with a certain "quality of service" (QoS). Here, network resources may be reserved for the unidirectional flow to ensure that the QoS associated with the data flow is maintained. The Resource ReSerVation Protocol (RSVP) is a network-control protocol that |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | enables applications to reserve resources in order to obtain special QoS for their data flows. RSVP works in conjunction with routing protocols to, e.g., reserve resources for a data flow in a computer network in order to establish a level of QoS required by the data flow. RSVP is defined in R. Braden, et al., Resource ReSerVation Protocol (RSVP), RFC 2205, the contents of which are hereby incorporated by reference in its entirety. In the case of traffic engineering applications, RSVP signaling (with Traffic Engineering extensions) is used to establish a TE-LSP and to convey various TE-LSP attributes to routers, such as border routers, along the TE-LSP obeying the set of required constraints whose path may have been computed by various means." Vasseur '879, 3:37-57.<br><br>"Generally, a tunnel is a logical structure that encapsulates a packet (a header and data) of one protocol inside a data field of another protocol packet with a new header. In this manner, the encapsulated data may be transmitted through networks that it would otherwise not be capable of traversing. More importantly, a tunnel creates a transparent virtual network link between two network nodes that is generally unaffected by physical network links or devices (i.e., the physical network links or devices merely forward the encapsulated packet based on the new header). While one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 3:58-4:3.<br><br>"Occasionally, a network element (e.g., a node or link) will fail, causing redirection of the traffic that originally traversed the failed network element to other network elements that bypass the failure. Generally, notice of this failure is relayed to the nodes in the network through an advertisement of the new network topology, e.g., an IGP or BGP Advertisement, and routing tables are updated to avoid the failure accordingly. Reconfiguring a network in response to a network element failure using, e.g., pure IP rerouting, can be time consuming. Many recovery techniques, however, are available to provide fast recovery and/or network configuration in the event of a network element failure, including, inter alia, "Fast Reroute", e.g., MPLS TE Fast Reroute. An example of MPLS TE Fast Reroute is described in Pan, et al., Fast Reroute Extensions to RSVP-TE for LSP Tunnels, RFC 4090, May 2005, which is hereby incorporated by reference as though fully set forth herein." Vasseur '879, 4:4-21. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "Fast Reroute (or FRR) has been widely deployed to protect against network element failures, where "backup tunnels" are created to bypass one or more protected network elements (e.g., links, shared risk link groups (SRLGs), and nodes). When the network element fails, traffic is quickly diverted ("Fast Rerouted") over a backup tunnel to bypass the failed element, or more particularly, in the case of MPLS, a set of primary TE-LSPs (tunnels) is quickly diverted. Specifically, the point of local repair (PLR) node configured to reroute the traffic inserts ("pushes") a new label for the backup tunnel, and the traffic is diverted accordingly. Once the failed element is bypassed, the backup tunnel label is removed ("popped"), and the traffic is routed along the original path according to the next label (e.g., that of the original TE-LSP). Notably, the backup tunnel, in addition to bypassing the failed element along a protected primary TE-LSP, also intersects the primary TE-LSP, i.e., it begins and ends at nodes along the protected primary TE-LSP." Vasseur '879, 4:22-39.<br><br>"To offer maximum protection, e.g., guaranteed bandwidth, during Fast Reroute, backup tunnels may reserve a configurable amount of bandwidth to ensure that Fast Rerouted traffic from the primary tunnel has a reserved path to follow. For example, the bandwidth reserved for the primary tunnel may also be reserved on the backup tunnel. While this approach provides maximum protection, it also requires a non-negligible amount of network resources (e.g., capacity/bandwidth) and may increase operational complexity." Vasseur '879, 4:40-48.<br><br>"Certain techniques are available to efficiently minimize the amount of resources required by the establishment and maintenance of the backup tunnels for Fast Reroute. One such technique is to create zero-bandwidth ("0-BW") backup tunnels (i.e., tunnels that reserve no bandwidth) to protect non-0-BW primary tunnels. This "best effort" approach does not guarantee that the path followed by the backup tunnel will have enough bandwidth to support the diverted primary tunnel at the time of failure without QoS degradation, however in many situations the path quality of the backup tunnel is sufficient. For instance, if the network is not overly congested, or the backup tunnel follows a non-congested path, there may be enough available bandwidth along the backup tunnel to support the newly rerouted traffic. Also, because primary tunnels often reserve bandwidth in response to "peak" traffic utilization, the amount of traffic over the primary tunnel at the time of failure may be far less than the reserved bandwidth (e.g., at "off-peak" times). Notably, those skilled in the art will understand that in |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | the absence of the above exceptions, a 0-BW backup tunnel may have unacceptable bandwidth (e.g., affecting path quality) to support the diverted traffic." Vasseur '879, 4:49-5:2.<br><br>"Currently, head-end nodes (LSRs) may be configured to systematically reroute the primary tunnels affected by the network element failure (e.g., diverted primary tunnels), especially in the case with 0-BW backup tunnels, such as, e.g., by reestablishing a new primary tunnel that follows a path excluding the failed network element. In particular, 0-BW backup tunnels represent a best effort attempt to allow the head-end node to more gracefully reestablish the primary tunnel in response to a failure, since the backup tunnels may not be able to support the diverted traffic without QoS degradation. The systematic reestablishing may potentially result in the reestablishment of a large number of primary tunnels (e.g., up to 3000 for a single network element failure in today's networks). Notably, reestablishing diverted primary tunnels may be undesirable for the network, such as by creating traffic churn, jitter, control plane overloads, etc., as will be understood by those skilled in the art. However, as noted above, there are situations where the backup tunnel may provide acceptable bandwidth, at least, for example, for a period of time (e.g., possibly short) until the failed network element is restored. In these situations, then, it may have been unnecessary to reestablish the diverted primary tunnels. There remains a need, therefore, for a technique that dynamically determines whether to reestablish a diverted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network." Vasseur '879, 5:3-28.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 5:32-47. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as Resource ReSerVation Protocol (RSVP) Traffic Engineering (TE) signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or type/length/value (TLV) encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object of the RSVP messages to convey the path quality notification." Vasseur '879, 5:48-58.<br><br>"In accordance with one aspect of the present invention, the head-end node requests the establishment of the primary tunnel (e.g., a TE-Label Switched Path, TE-LSP), along with a request for Fast Reroute protection of one or more network elements (e.g., with zero-bandwidth, 0-BW backup tunnels) at a PLR node. In addition, the head-end node may include a request for backup tunnel path quality notification, such as, e.g., through the use of the novel Feedback sub-object. The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. Once failure is detected, the PLR node diverts the traffic onto the backup tunnel, and sends an error message (e.g., an RSVP PathErr) to the head end node indicating the "Fast Rerouting" of the primary tunnel." Vasseur '879, 5:59-6:6.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, the PLR node may collect metrics/statistics (e.g., packet drops, path cost, jitter, etc.) of the primary and/or backup tunnels. Once the primary tunnel is Fast Rerouted, the PLR node continues to collect metrics of the backup tunnel, and may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. Notably, the PLR node may be configured to send path quality notifications to the head-end node once, continually, periodically, in response to configurable changes in path quality, etc. Also, as in the case where multiple primary tunnels are Fast Rerouted, the path quality notification may include an indication of which Fast Rerouted primary tunnels in particular have been effected by the changed path quality." Vasseur '879, 6:7-23. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "In accordance with yet another aspect of the present invention, upon receiving the error message (PathErr), the head-end node may wait for the path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. Notably, in the event the head-end node does not receive a path quality notification for the backup tunnel (e.g., within a configurable time limit), the head-end node may attempt to reestablish the new primary tunnel. Moreover, where the head-end node has multiple primary tunnels being Fast Rerouted, a configurable subset of the primary tunnels may be reestablished, e.g., to reduce congestion of the backup tunnels, and/or to limit the number of reestablished primary tunnels within a given period of time." Vasseur '879, 6:24-43.<br><br>"Advantageously, the novel technique dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. By providing the head-end node of the primary tunnel with path quality feedback of the backup tunnel, the novel technique avoids reestablishing a potentially large number of tunnels over one or more alternate paths after a failure (and Fast Reroute) if the backup tunnels have acceptable path quality. In particular, the backup tunnels, e.g., 0-BW backup tunnels, may not be congested or subsequently burdened by the Fast Rerouted traffic of the primary tunnel. Also, the failed network element (thus the primary tunnel) may be quickly restored; therefore by not reestablishing the primary tunnel, network jitter, churn, etc., may be avoided. Further, the dynamic nature of the novel technique alleviates the need for cumbersome manual configuration." Vasseur '879, 6:44-59.<br><br>"FIG. 3 is schematic block diagram of an exemplary signaling (RSVP) message that may be advantageously used with the present invention." Vasseur '879, 7:6-8.<br><br>"FIG. 5 is a schematic block diagram of the computer network in FIG. 1 showing Fast Reroute protection of a primary tunnel using a backup tunnel in accordance with the present invention." Vasseur '879, 7:12-15. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 6 is a schematic block diagram of the computer network in FIG. 5 showing an unacceptable backup tunnel path quality and resultant reestablishing of the primary tunnel in accordance with the present invention." Vasseur '879, 7:16-19.<br><br>"FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention." Vasseur '879, 7:20-24.<br><br>"FIG. 1 is a schematic block diagram of an exemplary computer network 100 comprising a plurality of nodes A-F, such as routers or other network devices, interconnected as shown. The nodes may be a part of one or more autonomous systems, routing domains, or other networks or subnetworks. For instance, routers A and E may be provider edge (PE) devices of a provider network, (e.g., a service provider network) that are interconnected to one or more customer networks through customer edge (CE) devices (not shown, while the remaining nodes B-D and F may be core provider (P) devices, as will be understood by those skilled in the art. Those skilled in the art will also understand that the nodes A-F may be any nodes within any arrangement of computer networks, and that the view shown herein is merely an example. For example, the nodes may be configured as connections to/from one or more virtual private networks (VPNs), as will be understood by those skilled in the art. These examples are merely representative. Those skilled in the art will understand that any number of routers, nodes, links, etc. may be used in the computer network 100 and connected in a variety of ways, and that the view shown herein is for simplicity." Vasseur '879, 7:29-49.<br><br>"Data packets may be exchanged among the computer network 100 using predefined network communication protocols such as the Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Asynchronous Transfer Mode (ATM) protocol, Frame Relay protocol, Internet Packet Exchange (IPX) protocol, etc. Routing information may be distributed among the routers of the computer network using predetermined Interior Gateway Protocols (IGPs), such as conventional distance-vector protocols or, illustratively, link-state protocols, through the use of IGP Advertisements." Vasseur '879, 7:50-60. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 2 is a schematic block diagram of an exemplary router 200 that may be advantageously used with the present invention, e.g., as an edge router or a core router. The router comprises a plurality of network interfaces 210, a processor 220, and a memory 240 interconnected by a system bus 250. The network interfaces 210 contain the mechanical, electrical and signaling circuitry for communicating data over physical links coupled to the network 100. The network interfaces may be configured to transmit and/or receive data using a variety of different communication protocols, including, inter alia, TCP/IP, UDP, ATM, synchronous optical networks (SONET), wireless protocols, Frame Relay, Ethernet, Fiber Distributed Data Interface (FDDI), etc." Vasseur '879, 7:61-8:6.<br><br>"The memory 240 comprises a plurality of storage locations that are addressable by the processor 220 and the network interfaces 210 for storing software programs and data structures associated with the present invention. The processor 220 may comprise necessary elements or logic adapted to execute the software programs and manipulate the data structures. A router operating system 242 (e.g., the Internetworking Operating System, or IOS™, of Cisco Systems, Inc.), portions of which is typically resident in memory 240 and executed by the processor, functionally organizes the router by, inter alia, invoking network operations in support of software processes and/or services executing on the router. These software processes and/or services may comprise routing services 247, Traffic Engineering (TE) services 244, and RSVP services 249. It will be apparent to those skilled in the art that other processor and memory means, including various computer-readable media, may be used to store and execute program instructions pertaining to the inventive technique described herein." Vasseur '879, 8:7-26.<br><br>"Routing services 247 contain computer executable instructions executed by processor 220 to perform functions provided by one or more routing protocols, such as IGP (e.g., OSPF and IS-IS), IP, BGP, etc. These functions may be configured to manage a forwarding information database (not shown) containing, e.g., data used to make forwarding decisions. Routing services 247 may also perform functions related to virtual routing protocols, such as maintaining VRF instances (not shown) as will be understood by those skilled in the art." Vasseur '879, 8:27-36. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "RSVP services 249 contain computer executable instructions for implementing RSVP and processing RSVP messages in accordance with the present invention. RSVP is described in RFC 2205, entitled Resource ReSerVation Protocol (RSVP), and in RFC 3209, entitled RSVP-TE: Extensions to RSVP for LSP Tunnels, both as incorporated above." Vasseur '879, 8:37-42.<br><br>"TE services 244 contain computer executable instructions for operating TE functions in accordance with the present invention. Examples of Traffic Engineering are described in RFC 3209, RFC 3784, and RFC 3630 as incorporated above, and in RFC 3473, entitled, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions dated January 2003, which is hereby incorporated by reference in its entirety. A TE database (TED, not shown) may be illustratively resident in memory 240 and used to store TE information provided by the routing protocols, such as IGP, BGP, and/or RSVP (with TE extensions, e.g., as described herein), including, inter alia, path quality information as described herein. The TED may be illustratively maintained and managed by TE services 244." Vasseur '879, 8:43-57.<br><br>"Changes in the network topology may be communicated among routers 200 using a link-state protocol, such as the conventional OSPF and IS-IS protocols. Suppose, for example, that a communication link fails or a cost value associated with a network node changes. Once the change in the network's state is detected by one of the routers, that router may flood an IGP Advertisement communicating the change to the other routers in the network. In this manner, each of the routers eventually "converges" to an identical view of the network topology." Vasseur '879, 8:58-67.<br><br>"In one embodiment, the routers described herein are IP routers that implement Multi-Protocol Label Switching (MPLS) and operate as label switched routers (LSRs). In one simple MPLS scenario, at an ingress to a network, a label is assigned to each incoming packet based on its forwarding equivalence class before forwarding the packet to a next-hop router. At each router, a forwarding selection and a new substitute label are determined by using the label found in the incoming packet as a reference to a label forwarding table that includes this information. At the network egress, a forwarding decision is made based on the incoming label |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | but optionally no label is included when the packet is sent on to the next hop. In some network configurations, one hop prior to the network egress, a penultimate hop popping (PHP) operation may be performed. Particularly, because the hop prior to the network egress (the penultimate hop) is attached to the network egress, the label is no longer needed to assure that the traffic follows a particular path to the network egress. As such, the PHP-enabled device "pops" the labels from the traffic before forwarding the traffic to the network egress, e.g., using conventional or native (IP) routing, thereby alleviating the task of removing the labels at the network egress." Vasseur '879, 9:1-23.<br><br>"The paths taken by packets that traverse the network in this manner are referred to as label switched paths (LSPs) or Traffic Engineering (TE)-LSPs. An example TE-LSP is shown as the thick line and arrow (T1) between a head-end node (router A) and a tailend node (router E) in FIG. 1. Establishment of a TE-LSP requires computation of a path, signaling along the path, and modification of forwarding tables along the path. MPLS TE establishes LSPs that have guaranteed bandwidth under certain conditions. Illustratively, the TE-LSPs may be signaled through the use of the RSVP protocol (with Traffic Engineering extensions), and in particular, RSVP TE signaling messages. Notably, when incorporating the use of PCEs (described below), the path computation request (and response) between PCC and PCE can be exchanged in accordance with a protocol specified in Vasseur, et al., Path Computation Element (PCE) Communication Protocol (PCEP)—Version 1—<draft-vasseur-pce-pcep-02.txt>, Internet Draft, September 2005, the contents of which are hereby incorporated by reference in its entirety. It should be understood that the use of RSVP or PCEP serves only as an example, and that other communication protocols may be used in accordance with the present invention." Vasseur '879, 9:24-45.<br><br>"In accordance with RSVP, to request a data flow (TE-LSP) between a sender and a receiver, the sender may send an RSVP path request (Path) message downstream to the receiver along a path (e.g., a unicast route) to identify the sender and indicate e.g., bandwidth needed to accommodate the data flow, along with other attributes of the TE-LSP. The Path message may contain various information about the data flow including, e.g., traffic characteristics of the data flow. Also in accordance with the RSVP, a receiver establishes the TE-LSP between the sender and receiver by responding to the sender's Path message with a reservation request |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | (Resv) message. The reservation request message travels upstream hop-by-hop along the flow from the receiver to the sender. The reservation request message contains information that is used by intermediate nodes along the flow to reserve resources for the data flow between the sender and the receiver, to confirm the attributes of the TE-LSP, and provide a TE-LSP label. If an intermediate node in the path between the sender and receiver acquires a Path message or Resv message for a new or established reservation (TE-LSP) and encounters an error (e.g., insufficient resources, failed network element, etc.), the intermediate node generates and forwards a path or reservation error (PathErr or ResvErr, hereinafter Error) message to the sender or receiver, respectively." Vasseur '879, 9:46-10:2.<br><br>"FIG. 3 is a schematic block diagram of portions of a signaling message 300 (e.g., RSVP message, such as Path, Resv or Error) that may be advantageously used with the present invention. Message 300 contains, inter alia, a common header 310 and one or more signaling protocol specific objects 320, such as an LSP-ATTRIBUTE object 330. The common header 310 may comprise a source address 312 and destination address 314, denoting the origination and requested termination of the message 300. Protocol specific objects 320 contain objects necessary for each type of message 300 (e.g., Path, Resv, Error, etc.). For instance, a Path message may have a sender template object, Tspec object, Previous-hop object, etc. The LSP-ATTRIBUTE object 330, for instance, may be used to signal attributes and/or information regarding an LSP (tunnel). To communicate this information, LSP-ATTRIBUTE object 330 (as well as specific objects 320) may include various type/length/value (TLV) encoding formats and/or flags, as will be understood by those skilled in the art. An example of an LSP-ATTRIBUTE object is further described in Farrel, et al., Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE <draft-ietf-mpls-rsvpte-attributes-05.txt>, Internet Draft, May 2005, which is hereby incorporated by reference as though fully set forth herein. A Resv message, on the other hand, may have specific objects 320 for a label object, session object, filter spec object, etc., in addition to the LSP-ATTRIBUTE object 330. Error messages 300 (e.g., PathErr or ResvErr), may also have specific objects 320, such as for defining the type of error, etc." Vasseur '879, 10:3-31. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "It should be noted that in accordance with RSVP signaling, the state of the TE-LSP is refreshed on a timed interval, e.g., every thirty seconds, in which RSVP Path and Resv messages are exchanged. This timed interval is configurable by a system administrator. Moreover, various methods understood by those skilled in the art may be utilized to protect against route record objects (RROs) contained in signaling messages for a TE-LSP in the event security/privacy is desired. Such RRO filtering prevents a head-end node of the TE-LSP from learning of the nodes along the TE-LSP, i.e., nodes within the provider network." Vasseur '879, 10:4-42.<br><br>"Although the illustrative embodiment described herein is directed to MPLS, it should also be noted that the present invention may advantageously apply to Generalized MPLS (GMPLS), which pertains not only to packet and cell-based networks, but also to Time Division Multiplexed (TDM) and optical networks. GMPLS is well known and described in RFC 3945, entitled Generalized Multi-Protocol Label Switching (GMPLS) Architecture, dated October 2004, and RFC 3946, entitled Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control, dated October 2004, the contents of both of which are hereby incorporated by reference in their entirety." Vasseur '879, 10:43-55.<br><br>"To obviate delays associated with updating routing tables when attempting to avoid a failed network element (i.e., during convergence), some networks have employed MPLS TE Fast Reroute (FRR). MPLS Fast Reroute is a technique that may be used to quickly divert ("Fast Reroute") traffic around failed network elements in a TE-LSP. MPLS Fast Reroute is further described, for example, by Fast Reroute Extensions to RSVP-TE for LSP Tunnels, as incorporated by reference above. According to the technique, one or more network elements (e.g. links or nodes) in a network are protected by backup tunnels following an alternate path. If a failure occurs on a protected link or node, TE-LSPs (and consequently the traffic that they carry) are locally diverted onto an appropriate alternate path (e.g., a "backup tunnel") by the node immediately upstream from the failure. The backup tunnel acts as a Fast Reroute path for the primary TE-LSP and obviates delays associated with other measures, such as tearing down the primary TE-LSP after having gracefully diverted the TE-LSPs affected by the failure, should an alternate path around the failed network element exist. In the event of a |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | failure of a protected element the head-end node of the backup tunnel (or a "point of local repair," PLR node) may quickly begin diverting traffic over the backup tunnel with minimal disruption to traffic flow. Those skilled in the art will understand that MPLS Fast Reroute is one example of link or node failure protection, and that other known correction mechanisms may be used in accordance with the present invention. As mentioned above, however, the head-end node of the Fast Rerouted primary tunnel may attempt to reestablish the primary tunnel in response to learning of the protected element failure, particularly in the case where the backup tunnel is a zero-bandwidth (0-BW) tunnel. The attempt to reestablish the primary tunnel has conventionally been a systematic response to Fast Rerouting (diverting) of the primary tunnel, regardless of the path quality of the backup tunnel." Vasseur '879, 10:56-11:23.<br><br>"The present invention is directed to a technique for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network. According to the novel technique, a head-end node establishes a primary tunnel to a destination, and a point of local repair (PLR) node along the primary tunnel establishes a backup tunnel around one or more protected network elements of the primary tunnel, e.g., for Fast Reroute protection. Once one of the protected network elements fail, the PLR node "Fast Reroutes," i.e., diverts, the traffic received on the primary tunnel onto the backup tunnel, and sends notification of backup tunnel path quality (e.g., with one or more metrics) to the head-end node. The head-end node then analyzes the path quality metrics of the backup tunnel to determine whether to utilize the backup tunnel or reestablish a new primary tunnel." Vasseur '879, 11:24-39.<br><br>"In the illustrative embodiment described herein, the notification of backup tunnel path quality may be embodied as extensions to a request/response signaling exchange, such as RSVP TE signaling messages. Notably, the RSVP extensions are, in turn, embodied as new RSVP objects, flags, and/or TLV encoded formats contained within the RSVP objects. For instance, a novel Fast Reroute Feedback (FFeed) sub-object may be included within an LSP-ATTRIBUTE object 330 of the RSVP messages 300 to convey the path quality notification." Vasseur '879, 11:40-49. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 4 is a schematic block diagram illustrating the format of an extension object (TLV) 400, such as a Fast Reroute Feedback object, that may be advantageously used with the present invention. The extension object (or sub-object) 400 is illustratively embodied as a TLV contained in an LSP-ATTRIBUTE object 330 of an RSVP message 300 and is extended to carry backup (and/or primary) tunnel path quality information. To that end, the "Feedback" object 400 is organized to include a Type field 405 containing a predetermined type value signifying the specific content of the object 400. The Length field 410 is a variable length value. The TLV encoded format may also comprise one or more non-ordered sub-TLVs 450 carried within the TLV "payload" (e.g. Value field 415), each having a Type field 455, Length field 460, and Value field 465. The fields of the TLV 400 and sub-TLV(s) 450 are used in a variety of manners, including as described herein, according to the present invention." Vasseur '879, 11:59-12:8.<br><br>"In accordance with one aspect of the present invention, the head-end node (e.g., router A) requests the establishment of the primary tunnel (e.g., T1), such as a TE-LSP. Notably, the head-end node may be a head-end node for multiple primary tunnels, as will be understood by those skilled in the art. Along with the primary tunnel establishment, the head-end node may also request Fast Reroute protection of one or more network elements (e.g., all intermediate network elements) at a PLR node (e.g., router B as shown). Note that each intermediate node along the primary tunnel may act as a PLR node, and that router B is shown merely for simplicity. Illustratively, the Fast Reroute protection may be embodied as one or more zero-bandwidth (0-BW) backup tunnels at the PLR node (e.g., BT1). Those skilled in the art will also understand that the PLR node may protect more than one primary tunnel originating at more than one corresponding head-end node (not shown)." Vasseur '879, 12:9-25.<br><br>"The primary and backup tunnels may then be established, and, in accordance with Fast Reroute, the PLR node may monitor the protected network elements for failure. For example, various connectivity verification protocols, such as, e.g., Bidirectional Forwarding Detection (BFD), IGP "Hello" packets, BGP KEEPALIVE messages, etc., may be used to detect a failure of a network element, as will be understood by those skilled in the art. Furthermore, other lower-layer failure detection mechanisms (e.g. optical or SONET/SDH alarms) may be used to detect a network element failure. Once failure is detected, the PLR node diverts the |

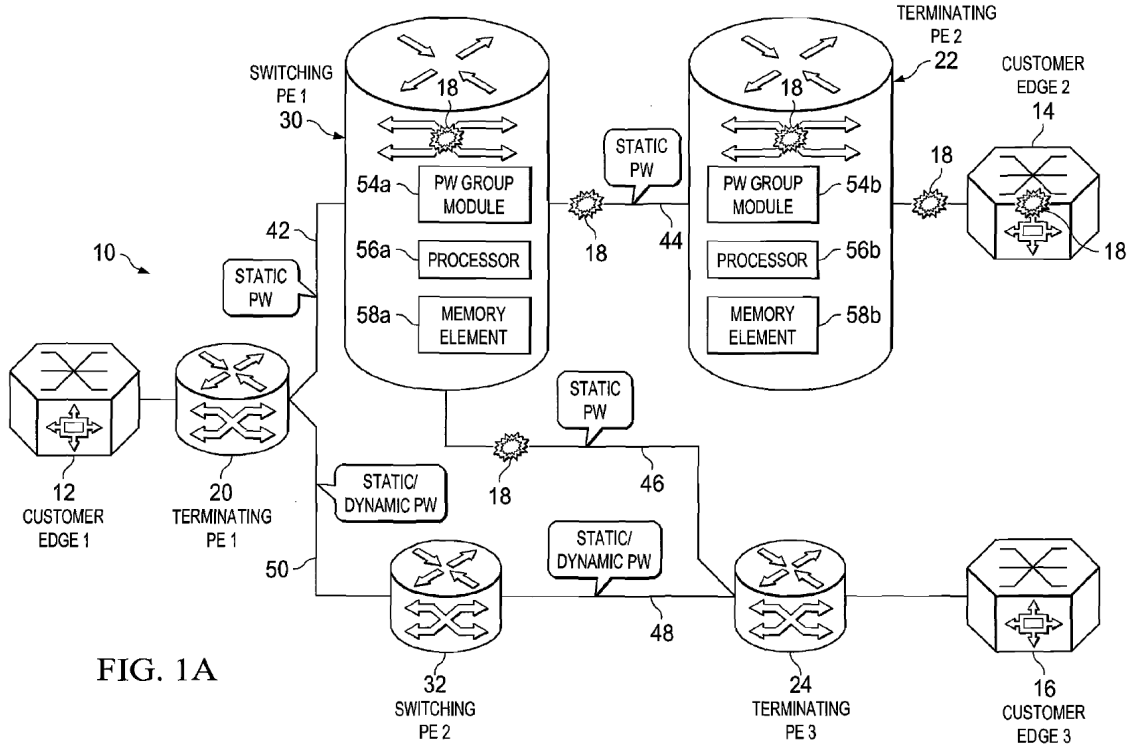
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | traffic onto the backup tunnel, and may send an error message (e.g., an RSVP PathErr 300, such as a conventional "tunnel locally repaired" message) to the head end node indicating the "Fast Rerouting" of the primary tunnel. FIG. 5 is a schematic block diagram of the computer network 100 in FIG. 1 showing Fast Reroute protection of the primary tunnel T1 (e.g., in response to a protected network element, router C, failure, indicated with an overlaid "X") using a backup tunnel BT1 in accordance with the present invention. Traffic originally received at the PLR node (router B) over the primary tunnel is now diverted over the backup tunnel to a remerge point (router D) of the primary tunnel, as will be understood by those skilled in the art." Vasseur '879, 12:42-65.<br><br>"In accordance with another aspect of the present invention, prior to Fast Rerouting, each PLR node may collect metrics/statistics of the primary and/or backup tunnels. For instance, example metrics may comprise, inter alia, packet drops, path cost, jitter, delay, bandwidth, etc. The PLR node may collect the metrics through traffic monitoring, probes, independent calculations, and/or through cooperation with protected nodes of the primary tunnel (primary nodes) and nodes of the backup tunnel (backup nodes), e.g., transmitting path quality notifications. Once the primary tunnel is Fast Rerouted (i.e., diverted after failure of a protected network element), the PLR node continues to collect metrics of the backup tunnel. (Alternatively, metrics of the backup tunnel may be collected only after Fast Reroute, and not prior to Fast Reroute.)" Vasseur '879, 12:66-13:12.<br><br>"For example, a path cost increase from the primary tunnel to the backup tunnel may be computed (and transmitted, below) by the PLR node prior to the failure (or during the failure while the primary TE-LSP is diverted onto the backup tunnel) using its own routing tables. The path cost increase may be calculated as a difference between the entire length (head-end node to tail-end node) of the primary and backup tunnels, or just the difference between the protected segment of the primary tunnel and the backup tunnel (PLR node to remerge point). Also, a jitter increase of the primary and backup tunnels, which may be generally described as a difference between inter-arrival of packets, may be monitored using various known techniques, such as, e.g., sending probe packets (probes) from the PLR node to the remerge point. For instance, probes may determine that packets arrive at the remerge point along the primary tunnel from the PLR node consistently, e.g., every 10 milliseconds (ms) (e.g., an |

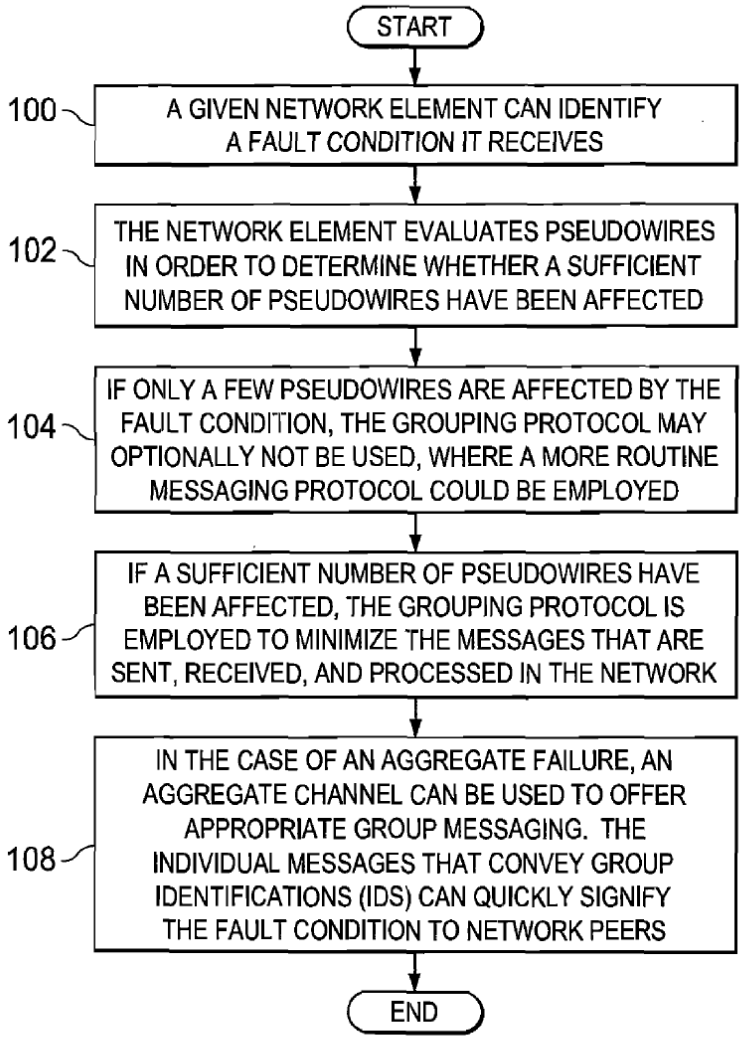
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | average value). After Fast Reroute, however, probes may determine that packets do not arrive at the remerge point along the backup tunnel from the PLR node consistently, e.g., one may arrive in 10 ms, another in 100 ms, another in 50 ms, etc. The non-constant rate of received packets (jitter) may be undesirable, e.g., in particular for voice over IP (VoIP) traffic, as will be understood by those skilled in the art." Vasseur '879, 13:13-36.<br><br>"As a further example, packet dropping may be monitored for the primary and backup tunnels prior to and after Fast Reroute (respectively). For instance, based on the tunnel label of the dropped packet, primary nodes and/or backup nodes may be able to distinguish which tunnel corresponds to the dropped packets. Each of the primary and/or backup nodes collect packet drop data, and periodically inform the PLR node of the number of dropped packets (e.g., though a corresponding Feedback object 400). In the case of a backup node, the PLR node receiving the notification may interpret the association of the backup tunnel label and the primary tunnel label to reference an appropriate primary tunnel. Those skilled in the art will understand that the above path quality metrics are merely examples, and that any other metrics/statistics useful for determining path quality of the backup tunnel may be used in accordance with the present invention (e.g., delay, bandwidth, etc.). Further, the path quality information may be measured and compared in a variety of manners, such as, e.g., as a difference between primary and backup tunnels before and after Fast Reroute, or simply the difference between the backup tunnel before and after Fast Reroute, etc." Vasseur '879, 13:37-58.<br><br>"Also after the primary tunnel is Fast Rerouted, the PLR node may inform the head-end node of the primary tunnel of any configurable difference (e.g., decrease) in path quality associated with utilizing the backup tunnel, i.e., in a path quality notification. For instance, the novel Feedback object 400 may include one or more sub-TLVs 450 corresponding to metrics/statistics, as described above. Notably, the path quality information pertaining to a particular metric/statistic may be transmitted as total values for interpretation by the head-end node (e.g., to determine the difference), or as PLR-node-computed differences (e.g., between the primary and backup tunnels, or before and after Fast Reroute). For instance, if the delay of the primary tunnel (along the protected segment) prior to Fast Reroute were 2 ms, and after Fast Reroute the delay of the backup tunnel were 5 ms, the notification may be configured to |

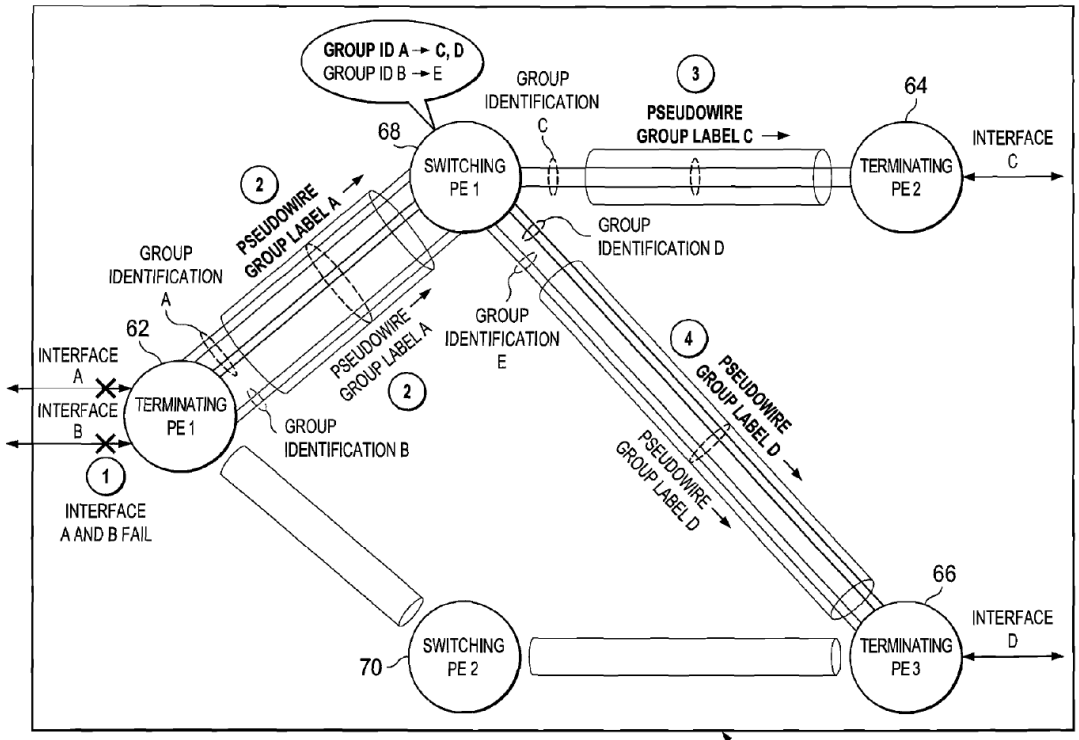
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | include both values 2 ms and 5 ms, or instead simply the difference, i.e., an increase of 3 ms." Vasseur '879, 13:59-14:8.<br><br>"In accordance with yet another aspect of the present invention, upon receiving the error message 300 (PathErr), the head-end node may wait for at least one path quality notification (i.e., if requested) prior to determining whether to reestablish the new primary tunnel. The determination may be made based on configurable boundaries, increases, combinations, etc., of the metrics. Also, any number of metrics may be used in the determination, e.g., as configured by a system administrator. For example, using the metrics described above, a head-end node may be configured to reestablish the primary tunnel in response to i) a certain number of packet drops, ii) a percent increase in packet drops, iii) a number of packet drops and a percent increase in path cost, iv) a percent increase in path cost and a percent increase in jitter, etc. Those skilled in the art will understand that these are merely examples of possible path quality comparisons and reestablishment determinations, and that any comparisons to any metrics at any configurable changes may be used in accordance with the present invention." Vasseur '879, 14:41-59.<br><br>"If the metrics are acceptable, the backup tunnel remains utilized and no primary tunnel reestablishment is performed. On the other hand, if the metrics are unacceptable, the head-end node may attempt to reestablish the new primary tunnel. FIG. 6 is a schematic block diagram of the computer network 100 in FIG. 5 showing an unacceptable backup tunnel path quality (dotted line and arrow) and resultant reestablishment of the primary tunnel in accordance with the present invention. Those skilled in the art will understand that the reestablished primary tunnel may traverse one or more primary nodes (not shown), and that it may be computed specifically to avoid the failed network element and any network elements of the unacceptable backup tunnel. Those skilled in the art will also understand that the attempt to reestablish the primary tunnel may not be able to find an acceptable path, in which case the head-end node may continue to use the unacceptable backup tunnel or other unacceptable rerouted path." Vasseur '879, 14:60-15:9. |

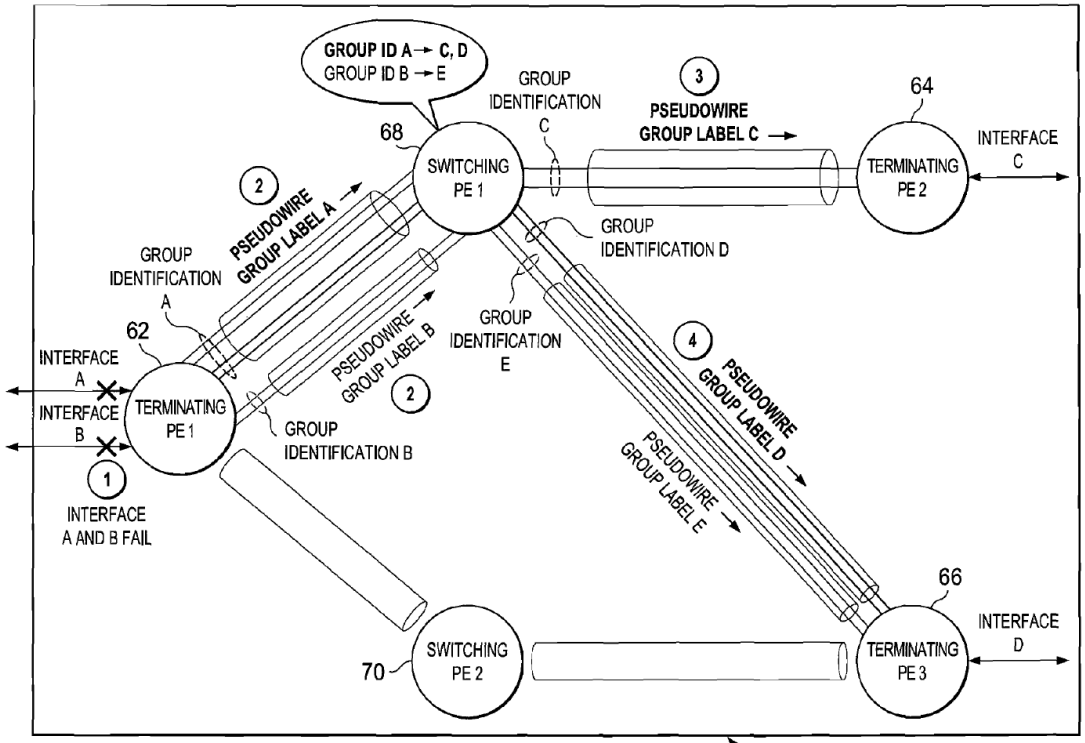
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIGS. 7A and 7B are flowcharts illustrating a procedure for dynamically determining whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in accordance with the present invention. The procedure 700 starts at step 705, and continues to step 710, where a head-end node (e.g., router A) establishes a primary tunnel (e.g., T1) to a destination tail-end node (e.g., router E). During or after establishment, the head-end node requests protection of the primary tunnel at step 715, and in response, PLR nodes along the primary tunnel (e.g., router B) create backup tunnels (e.g., BT1) to protect the primary tunnel in step 720. (Those skilled in the art will understand that backup tunnels around the protected network elements may already exist at the PLR node, and that "creating backup tunnels" in step 720 implies an association with pre-existing backup tunnels.) As mentioned above, these backup tunnels may illustratively be embodied as 0-BW backup tunnels. In accordance with the present invention, the head-end node may additionally request backup tunnel path quality notification from the PLR nodes in step 725, such as, e.g., through the use of empty corresponding Feedback objects in RSVP (Path) messages 300, as described above." Vasseur '879, 15:37-58.<br><br>"The procedure 700 continues to FIG. 7B (step "A"), where in step 750 the PLR node detecting the failure diverts ("Fast Reroutes") the primary tunnel traffic to the backup tunnel and sends an error message (e.g., an RSVP (Error) message 300) to the head end node, e.g., a "tunnel locally repaired" message. The detecting PLR node continues to collect path quality information for the backup tunnel in step 755 and at step 760 determines whether to send the path quality notification to the head-end node. For example, as mentioned above, the PLR node may be configured to continually send notifications, or periodically, or in response to a configurable change in path quality, etc. Also as mentioned above, the PLR node may be configured to send either the actual path quality information or the change (difference) in path quality. (As further mentioned above, metrics of the backup tunnel may alternatively be collected only after Fast Reroute, and not prior to Fast Reroute.)." Vasseur '879, 16:4-20.<br><br>"If the PLR node decides to send the notification in step 760, then the head-end node determines whether to reestablish the primary tunnel based on the backup tunnel path quality notification in step 765, e.g., based on one or more configurable thresholds, percentages, etc., as described above. If the backup tunnel is currently maintaining an acceptable quality for the |

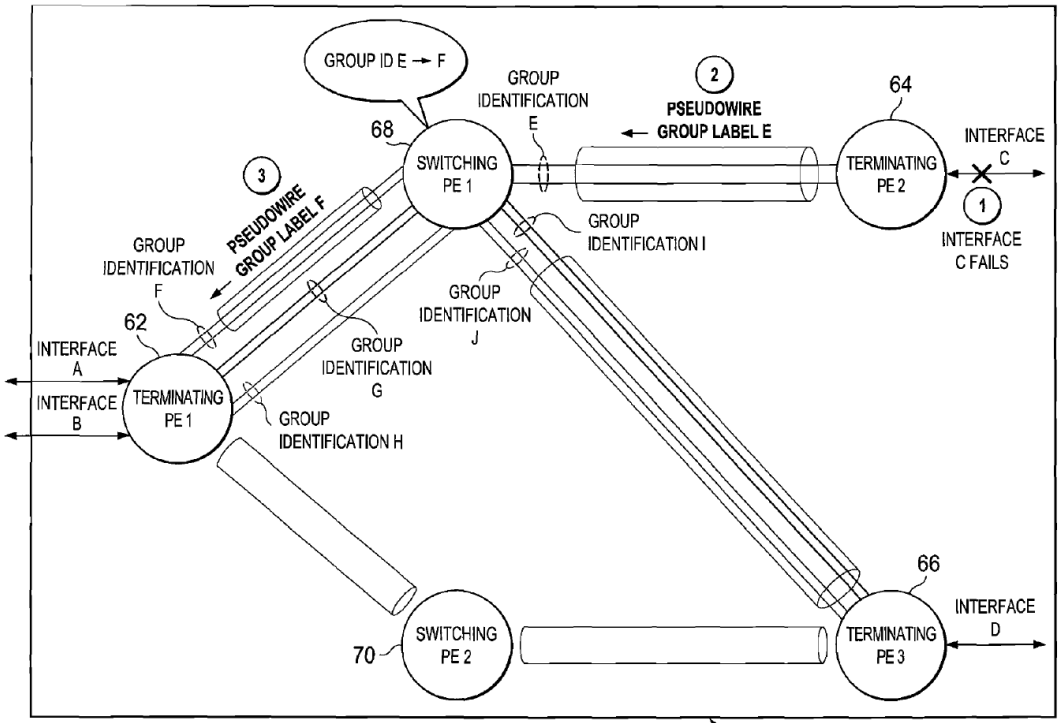
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | traffic flow in step 770, the head-end node may continue to utilize the backup tunnel, and the PLR node continues to collect path quality information in step 755 to detect any change in quality. Otherwise, if the backup tunnel quality is not acceptable in step 770, the head-end node may attempt to reestablish the primary tunnel in step 775. Notably, as mentioned above, if the PLR node has not sent any notification (step 760) within a configurable period of time in step 780, e.g., due to a backup tunnel failure, over-congestion, etc., then the head-end node may also attempt to reestablish the primary tunnel in step 775 accordingly. Moreover, as described above, in the event more than one primary tunnel is Fast Rerouted for the head-end node, various techniques to reestablish one or more of the primary tunnels may be used (e.g., as many tunnels as necessary, a configurable subset of tunnels, all tunnels, the congested tunnels, etc.). The procedure 700 ends in step 785." Vasseur '879, 16:21-43.<br><br>"While there has been shown and described an illustrative embodiment that dynamically determines whether to reestablish a Fast Rerouted primary tunnel based on path quality feedback of a utilized backup tunnel in a computer network, it is to be understood that various other adaptations and modifications may be made within the spirit and scope of the present invention. For example, the invention has been shown and described herein using "Fast Reroute," e.g., MPLS TE Fast Reroute (FRR). However, the invention in its broader sense is not so limited, and may, in fact, be used with other network element protection and failure correction mechanisms as will be understood by those skilled in the art. Moreover, while the above description describes performing the technique at the head-end node and PLR node, the invention may also be advantageously used with PCEs. In addition, while one example of a tunnel is an MPLS TE-LSP, other known tunneling methods include, inter alia, the Layer Two Tunnel Protocol (L2TP), the Point-to-Point Tunneling Protocol (PPTP), and IP tunnels." Vasseur '879, 16:63-17:13.<br><br>**Rustogi discloses**:<br>"An example method includes identifying a fault condition in a network, and evaluating pseudowires affected by the fault condition in order to make a determination as to whether an aggregate failure occurred in the network for a group of pseudowires. The method also includes communicating a group message indicating that the group of pseudowires is associated with the fault condition. The group message includes a group identification (ID), |

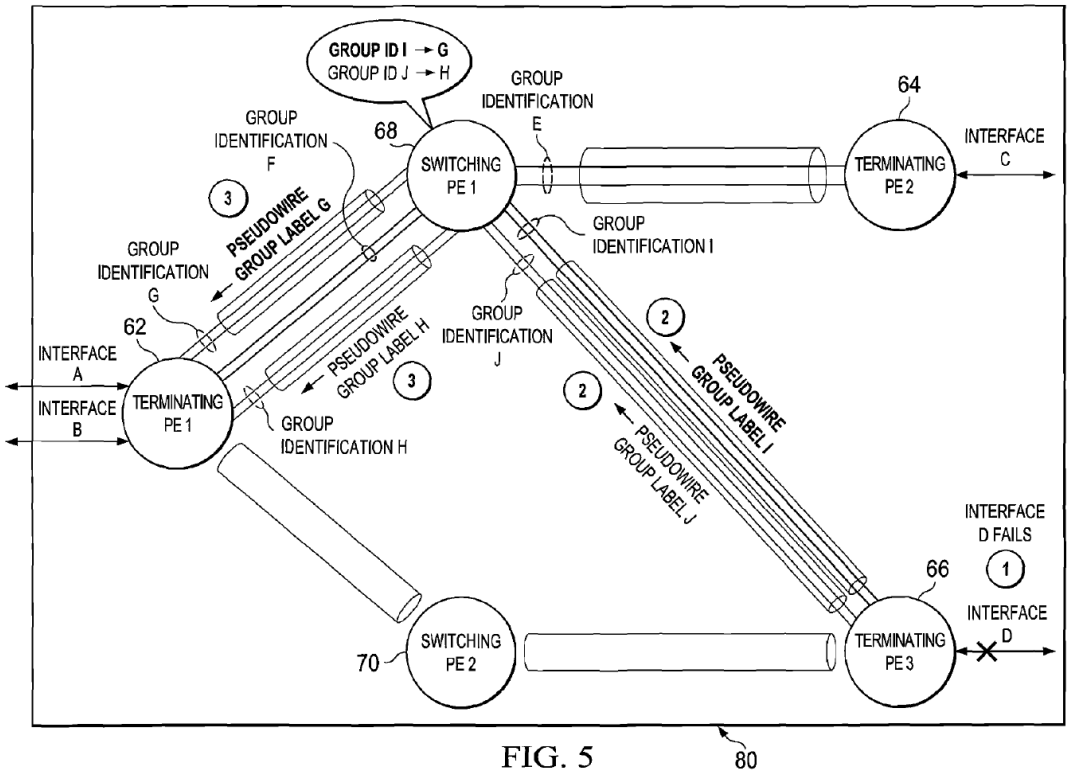
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | which identifies the group of pseudowires, and the group message includes a pseudowire group label identifying an in-band aggregate channel. More specifically, the pseudowire group label can be applicable to static pseudowires. In more detailed embodiments, the group ID identifies the group of pseudowires that are associated with an attachment circuit, a label switched path, or a port. Internal mappings can be maintained such that a plurality of pseudowires is mapped to individual interfaces of network elements in the network." Rustogi, Abstract.<br>  <br><br>FIG. 1A<br>Rustogi, FIG. 1A. |

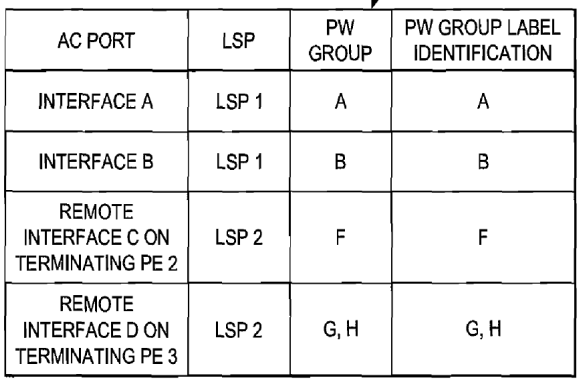
| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 1B<br>Rustogi, FIG. 1B. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Rustogi, FIG. 2. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Rustogi, FIG. 3. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Rustogi, FIG. 4. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 5<br>Rustogi, FIG. 5. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | 74<br>  <br>FIG. 6<br>Rustogi, FIG. 6.<br><br>"The field of communications has become increasingly important in today's society. In particular, the ability to quickly and to effectively provision connections presents a significant challenge to component manufacturers, system designers, and network operators. Multiprotocol Label Switching (MPLS) is a mechanism in telecommunications networks that carries data from one network node to the next. Layer 2 services (such as Frame Relay, Asynchronous Transfer Mode, and Ethernet) can be emulated over an MPLS backbone by encapsulating the Layer 2 Protocol Data Units (PDUs) and transmitting them over pseudowires. Protocols exist for establishing and maintaining the pseudowires. Certain issues have arisen in pseudowire scenarios, where faults are detected in the network." Rustogi, ¶ [0002].<br><br>"FIG. 1A is a simplified block diagram of a communication system for providing pseudowire group labels in a network environment in accordance with one embodiment of the present disclosure." Rustogi, ¶ [0004]. |

| AC PORT | LSP | PW GROUP | PW GROUP LABEL IDENTIFICATION |
|---|---|---|---|
| INTERFACE A | LSP 1 | A | A |
| INTERFACE B | LSP 1 | B | B |
| REMOTE INTERFACE C ON TERMINATING PE 2 | LSP 2 | F | F |
| REMOTE INTERFACE D ON TERMINATING PE 3 | LSP 2 | G, H | G, H |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 1B is a simplified flowchart depicting one possible, generic operational flow associated with the communication system." Rustogi, ¶ [0005].<br><br>"FIG. 2 is a simplified block diagram of an example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0006].<br><br>"FIG. 3 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0007].<br><br>"FIG. 4 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0008].<br><br>"FIG. 5 is a simplified block diagram of another example group labeling operation in accordance with one embodiment." Rustogi, ¶ [0009].<br><br>"FIG. 6 is a simplified table of an example set of pseudowire group provisioning parameters in accordance with one embodiment." Rustogi, ¶ [0010].<br><br>"FIG. 1A is a simplified block diagram of a communication system 10 for providing pseudowire group labels in accordance with one example of the present disclosure. FIG. 1A includes a customer edge 1 (CE1) 12, a CE2 14, and a CE3 16, where a number of faults 18 are shown as propagating in the network. Typically, when an error or a failure occurs in the network (e.g., an interface failure, a pulled cable, a switch failure, hardware/software failures generally, etc.), messages are sent to various network devices in order to inform them of these fault conditions. Faults 18 of FIG. 1A are indicative of such messages, where the underlying fault condition (being signaled by the messages) can occur virtually anywhere in a network (e.g., in a customer edge, in provider equipment, etc.). FIG. 1A also includes a terminating provider equipment 1 (TPE1) 20, a TPE2 22, a TPE3 24, a switching provider edge 1 (SPE1) 30, and a SPE2 32. In one particular example implementation, the TPEs and SPEs of FIG. 1A are switches that are configured to exchange data in a network environment." Rustogi, ¶ [0012]. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "SPE1 30 may include a pseudowire (PW) group module 54 a, a processor 56 a, and a memory element 58 a. In a similar fashion, TPE2 22 may include a pseudowire group module 54 b, a processor 56 b, and a memory element 58 b. FIG. 1A also includes a number of static pseudowires 42, 44, and 46. A set of static/dynamic pseudowires 48, 50 is also provided. Note that the group labeling protocol discussed herein can be executed between individual SPEs, TPEs, or between any combinations of these elements." Rustogi, ¶ [0013].<br><br>"In one particular arrangement, communication system 10 is provided in conjunction with a Layer-2 virtual private networks (L2VPN)/operation, administration, and maintenance (OAM) L2VPN/OAM framework. The OAM framework is intended to provide OAM layering across L2VPN services, pseudowires, and packet switched network (PSN) tunnels. Communication system 10 may include any suitable networking protocol or arrangement that provides a communicative platform for communication system 10. Thus, communication system 10 may include a configuration capable of transmission control protocol/internet protocol (TCP/IP) communications for the transmission and/or reception of packets in a network. Communication system 10 may also operate in conjunction with a user datagram protocol/IP (UDP/IP) or any other suitable protocol where appropriate and based on particular needs." Rustogi, ¶ [0014].<br><br>"Failure detection and failure notification for static pseudowires is inadequate, where sluggish signaling can result in poor scalability for failover performance. Typically, static pseudowires are manually configured at respective endpoints, where control channels are absent for providing group level signaling messages. Aggregate channels are significant tools for providing suitable scalability in the network, but no such aggregate channel exists for static pseudowires. For dynamic pseudowires, such an aggregate channel may be present in the form of a label distribution protocol (LDP) directed session. However, no such protocol exists for static pseudowire configurations such that an in-band aggregate channel would be available for static pseudowires." Rustogi, ¶ [0016]. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "Communication system 10 can address the aforementioned issues (and others) by offering a pseudowire group label that can represent an aggregate channel for groups of static pseudowires. The aggregate channel of communication system 10 can allow for improved scalability of failover performance. In accordance with one potential configuration of communication system 10, a pseudowire group label is representative of a group of static pseudowires transported over a label switched path (LSP). The pseudowire group label can identify the aggregate channel, which captures the hierarchy relevant to OAM mechanisms. Additionally, the groups represented by the group identification (ID) can be mutually exclusive, where a pseudowire is part of only one group. In other embodiments, a pseudowire can be part of multiple groups, or be configured in any other suitable manner based on particular network arrangements." Rustogi, ¶ [0017].<br><br>"During operations, and with brief reference to FIG. 1B, a given network element can identify a fault condition it receives (at step 100) and, subsequently, evaluate pseudowires in order to determine whether a sufficient number of pseudowires have been affected. This is reflected by step 102. If only a few pseudowires are affected by the fault condition, the grouping protocol outlined herein may have only nominal value, where there could be an option to simply communicate the fault condition in a more routine manner, as outlined in step 104. However, if a sufficient number of pseudowires have been affected, the grouping protocol outlined herein can be employed to minimize the messages that are sent, received, and processed in the network. This is reflected as step 106. Note that the determination (as to whether a sufficient number of pseudowires have been impacted by the fault condition) can involve accessing internal tables such that a quick mapping can occur to determine if an aggregate failure has occurred. As used herein, the term 'aggregate failure' simply connotes that a sufficient number of pseudowires have experienced the fault condition such that an aggregate channel can be employed to offer appropriate group messaging. For the aggregate failure condition, the individual messages that convey Group identifications (IDs) can quickly signify the fault condition to network peers, as shown in step 108." Rustogi, ¶ [0018]. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "In specific regards to OAM mechanisms, OAM messages typically result from common failures in the network. These fault conditions can be aggregated such that they are signaled as a single message, which could represent a group of failed pseudowires (as opposed to sending individual messages for each failed pseudowire). Hence, a single message could be sent to represent all the relevant OAM messages propagating in communication system 10. The group label that propagates in communication system 10 provides an architecture with a significant level of aggregation for failed pseudowires (i.e., pseudowires being affected by a given fault condition), particularly for OAM messaging. Moreover, the in-band aggregate channel of communication system 10 is based (at least in part) on the evolving trends of OAM mechanisms, which are required to be fast, responsive, and capable of being implemented in hardware or software. Additionally, in-band OAM protocols are a better measure of the path availability." Rustogi, ¶ [0020].<br><br>"In operation of one example implementation, a group label can represent the tuple <attachment circuit (AC) port level grouping, LSP>. This could signify that all pseudowires on an AC port (sought for aggregation) traverse a given LSP. Multiple pseudowire groups can exist within an LSP. Similarly, pseudowires on the same AC port (that traverse a different LSP) can use a different pseudowire group label. Alternatively, an administrator may seek to employ a one-to-one mapping between an LSP and a group label. If that were the case, then only one pseudowire group would exist within an LSP. In scenarios where there is no LSP label in the packet (e.g., due to penultimate hop popping), the pseudowire group label can provide the hierarchy that is appropriate." Rustogi, ¶ [0021].<br><br>"In one particular example, the group level pseudowire OAM message can be sent with the following label stack: Explicit/Implicit LSP Label+pseudowire group Label+GAL+ACH+pseudowire OAM with grouping TLV (where GAL=Generic Associated Channel Label, ACH=Associated Channel Header, TLV=Type-Length-Value). If there are multiple LSPs, then one group label can be provisioned for each LSP (for each pseudowire group), where per group messages can be sent on each LSP. The group label does not necessarily have a one-to-one mapping to the grouping of pseudowires implied by the Group ID in the grouping TLV. Note also that the group-based aggregate channel is applicable to |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | static pseudowires, as well as for dynamic pseudowires in certain applications." Rustogi, ¶ [0022].<br><br>"As discussed herein, the aggregate channel of communication system 10 can be configured in various ways. For example, and with regards to a first option, a separate label may simply be used to identify a pseudowire group within an LSP. The association of an OAM message and a pseudowire group is straightforward. There could potentially be multiple pseudowire group labels per LSP. As a second option, one group label can be used to identify a common pseudowire group channel on the LSP. In this implementation, one pseudowire group label is provided per LSP. The OAM message association to a pseudowire group is not as simple as the first option. As a third option, one pseudowire is simply designated to convey grouping information (e.g., without using a group label). In this case, there is no need for a pseudowire group label. Again, the OAM message association to a pseudowire group is not as simple as the first option." Rustogi, ¶ [0023].<br><br>"Any combination of formatting (for the Group ID and the pseudowire group label) can be used in the group message to be communicated in the network. In one example, only one of these elements is communicated when an aggregate fault condition is detected, or these elements can be combined into a single unique identifier. In the most generic example, a group message would at least include the Group ID (identifying the pseudowires affected by the fault) and a pseudowire group label (identifying an aggregate channel for communicating the group message). In this generic sense, a pipe (the Group ID) within a pipe (the pseudowire group label) is being identified, where the group message is identifying both elements during an aggregate fault condition. Operational details of communication system 10 are described below with reference to FIGS. 2-6. Note that before turning to additional example flows and example embodiments of the present disclosure, a brief overview of the infrastructure of communication system 10 is provided." Rustogi, ¶ [0024].<br><br>"CE1 12, CE2 14, and CE3 16 represent devices, infrastructure, equipment, clients, or customers seeking to initiate a data session in communication system 10. These elements may can comprise a digital subscriber line access multiplexer (DSLAM), a router, a personal computer, a server, a switch, and/or other devices associated with data propagation. Further, |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | these elements may sit behind, or in front of, one or more of these identified devices. The term 'CE' may be inclusive of the devices identified above (e.g., a DSLAM, a switch, etc.), as well as devices used to initiate a communication, such as a console, a proprietary endpoint, a telephone, a cellular telephone, a bridge, a computer, a personal digital assistant (PDA), a laptop or an electronic notebook, or any other device, component, element, or object capable of initiating voice, audio, media, or data exchanges within communication system 10. The customer element may also include any device that seeks to initiate a communication on behalf of another entity or element, such as a program, a database, or any other component, device, element, or object capable of initiating a voice, a video, text, or a data exchange within communication system 10. Data, as used herein in this document, refers to any type of video, numeric, voice, media, or script data, or any type of source or object code, or any other suitable information in any appropriate format that may be communicated from one point to another." Rustogi, ¶ [0025].<br><br>"SPE1 30, SPE2 32, TPE1 20, TPE2 22, and TPE3 24 are network elements that facilitate communications in two directions in a network environment. In one particular example, each of these network elements is a switch configured to exchange data over static and/or dynamic pseudowire links. Further, the traffic exchanged between these components may be directed over an MPLS transport in certain embodiments. As used herein in this Specification, the term 'network element' is meant to encompass switches, routers, bridges, gateways, servers, processors, loadbalancers, firewalls, or any other suitable device, component, element, or object operable to exchange or process information in a network environment. Moreover, these network elements may include any suitable hardware, software, components, modules, interfaces, or objects that facilitate the operations thereof. This may be inclusive of appropriate algorithms and communication protocols that allow for the effective exchange of data or information. Along similar design alternatives, any of the internal modules and components of these network elements may be combined in various possible configurations." Rustogi, ¶ [0029]. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "Turning to FIG. 2, FIG. 2 is a simplified block diagram of an example system 60 for providing an example use case using per-label switched path (LSP) pseudowire group labels. FIG. 2 includes a TPE1 62, a TPE2 64, a TPE3 66, a SPE1 68, and a SPE2 70. Each pseudowire group is identified, where a group identification (ID) for Group A and Group B is depicted at TPE1 62. Similarly, Groups C, D, and E have Group IDs at SPE1 68. TPE2 64 and TPE3 66 can couple to interfaces C and D, respectively." Rustogi, ¶ [0031].<br><br>"In this particular example, interfaces A and B have failed. Note that there is a multitude of attachment circuits (e.g., 1000 attachment circuits) that are being transported over these interfaces A and B, where the attachment circuits are being tunneled into a corresponding number of pseudowires. For example, there could be 500 attachment circuits on interface A (implicating 500 pseudowires) and 500 attachment circuits on interface B, where the fault condition for the pseudowires should be signaled. In other flawed systems, an architecture would individually signal this fault condition for each pseudowire (e.g., via signaling between TPE1 62 and SPE1 68). Instead of sending 500 messages, a single message can be sent, where a single label (and Group ID) can be used to identify the pseudowires. In this case, the Group ID A is used to signal the fault condition for 300 pseudowires and for 200 pseudowires (i.e., the top two links connecting TPE1 62 and SPE1 68) using a single message (that includes Group Label A and Group ID A). Thus, the status for Group A is quickly communicated to SPE1 68. Similarly, Group ID B can be used to signal the status of the other 500 pseudowires to appropriately convey the status for Group B. More specifically, the message can include Group Label A and Group ID B. Note that all 1000 pseudowires have effectively been accounted for using these Group IDs A and B." Rustogi, ¶ [0032].<br><br>"FIG. 3 is a simplified block diagram of an example system 72 for providing another use case for pseudowire group labels. Note that the grouping mechanism outlined herein is not limited to pseudowires that propagate over LSPs. Certain pseudowires can propagate over an LSP and represent one group, where two ports can be provisioned for two different groups (e.g., Group A and Group B). Hence, FIG. 3 is depicting a use case using pseudowire group labels for <port, LSP>mapping. In a general sense, such a configuration is showing how pseudowire mechanics can be used to offer different group signaling, which may be based on various possible implementations. Thus, there is a group level construct corresponding to the group |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | labels that are created such that any OAM protocol can send the appropriate aggregate messages. In this particular example, the signaling for Group ID A, B, C, and D is similar to that of FIG. 2; however, the grouping mechanism has simply changed." Rustogi, ¶ [0035].<br><br>"FIG. 4 is a simplified block diagram of an example system 76 for providing another use case for pseudowire group labels. In this particular example, interface C fails (as shown at TPE2 64). Note that the same logical flow occurs in FIG. 4 in terms of the group signaling, as previously discussed. The group labels in two directions do not have to be the same, where the groupings for the messaging are not necessarily symmetrical. In this particular example, TPE2 64 sends a status for Group E with the corresponding group label (i.e., Group ID E for 300 pseudowires), where that message will have a Group Label E and a Group ID E. Hence, this particular signaling is indicative of 300 pseudowires failing in the network. SPE1 68 can send the status for Group F (where the Group ID F is associated with 300 pseudowires) to TPE1 62, where that message includes a Group Label F and a Group ID F." Rustogi, ¶ [0036].<br><br>"FIG. 5 is a simplified block diagram of an example system 80 for providing another use case for pseudowire group labels. In this particular example, interface D fails (as shown at TPE3 66), where all 700 pseudowires fail. In one implementation, TPE3 66 does not have a 700 pseudowire Group ID. Instead, the Group IDs can correspond to 200 and 500 pseudowires, when summed together account for the 700 pseudowires. In this particular example, TPE3 66 sends one message for Group I (representing 200 pseudowires) and another message for Group J (representing 500 pseudowires) to SPE1 68. In response, SPE1 68 sends a message for Group G (representing 200 pseudowires) and another message for Group H (representing 500 pseudowires). Again, the signaling being exchanged between these elements is minimal due to the effective grouping of pseudowires. SPE1 68 also sends a single message for Group I (associated with 200 pseudowires) and Group J (associated with 500 pseudowires) to TPE3 66, which is coupled to interface D. Group ID G is associated with 200 pseudowires, whereas Group ID H is representative of 500 pseudowires." Rustogi, ¶ [0037]. |

| No. | '821 Patent Claim 20 | The Reference |
|---|---|---|
| | | "FIG. 6 is a simplified table 74 illustrating an example set of pseudowire group provisioning parameters for TPE1 62, where these particular provisioning parameters could be relevant to the configuration of FIG. 3. At least in one generic sense, FIG. 2 can reflect one approach for mapping a PW group label to a PW Group ID, while FIGS. 3-5 can reflect a second approach for such mappings, where table 74 is associated with that second approach." Rustogi, ¶ [0038].<br><br>"In particular, table 74 illustrates the mapping between SPE1 68 and TPE1 62. The first column represents the attachment circuit port (e.g., interface A, interface B, remote interface C on TPE2 64, and remote interface D on TPE3 66). Additionally, table 74 depicts a number of LSPs, a set of pseudowire grouping labels, and a set of pseudowire Group IDs. Note that the Group IDs are provided inside the pseudowire group labels in this example such that these two columns match in table 74. Additionally, note that table 74 is merely representing some of the possible characteristics in a single direction, where different constructs could be used in the reverse direction. Note that the provisioning as discussed herein can significantly reduce messaging such that these presented concepts offer increased scalability. This is due in part to the nominal processing that occurs in the network, in contrast to the processing required to evaluate a prolific amount of signaling messages associated with particular pseudowires. Additionally, the paradigm discussed herein can afford service providers an adequate amount of downtime after a failure has occurred in the network." Rustogi, ¶ [0039]. |

**EXHIBIT E-3**
Defendant's First Amended Invalidity Contentions
*Orckit Corporation v. Cisco Systems, Inc.*, 2:22-cv-00276-JRG-RSP

---

**Chart for U.S. Patent 7,545,740 ("the '740 Patent")**
**35 U.S.C. §103**

In this chart, "Reference" refers to each of the following references:

- U.S. Patent Publication No. 2006/0221974 to Hilla et al. ("Hilla")
- U.S. Patent Publication No. 2003/0147387 to Devi et al. ("Devi")
- Cisco EtherChannel Implementation In Cisco Products ("Cisco EtherChannel System")
- IEEE Standard 802.3, 2002 ("IEEE 802.3")
- Cisco EtherSwitch System ("Cisco EtherSwitch System")
- U.S. Patent Publication No. 2004/0228278 to Bruckman et al. ("Bruckman")
- U.S. Patent Publication No. 2003/0210688 to Basso et al. ("Basso")
- U.S. Patent Publication No. 2006/0039366 to Ghosh et al. ("Ghosh")
- U.S. Patent Publication No. 2004/0042448 to Lebizay et al. ("Lebizay")
- U.S. Patent No. 6,081,530 to Wiher et al. ("Wiher '530")

The following additional references are identified individually:

- U.S. Patent Publication No. 2004/0037278 to Wong et al. ("Wong")
- U.S. Patent Publication No. 2004/0208197 to Viswanathan ("Viswanathan")
- U.S. Patent Publication No. 2006/0251074 to Solomon ("Solomon")
- U.S. Patent Publication No. 7,221,652 to Singh et al. ("Singh")
- U.S. Patent No. 8,990,430 to Smith et al. ("Smith '430")
- U.S. Patent No. 8,526,427 to Smith et al. ("Smith '427")
- U.S. Patent Publication No. 2006/0039384 to Dontu et al. ("Dontu")
- U.S. Patent No. 6,553,029 to Alexander ("Alexander '029")
- U.S. Patent No. 6,473,424 to DeJager et al. ("DeJager '424")
- U.S. Patent No. 7,554,914 to Li et al. ("Li '914")

- U.S. Patent No. 8,155,125 to Borgione et al. ("Borgione '125")

As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under 35 U.S.C. § 103 because The Reference renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's First Amended Invalidity Contentions.

Motivations to combine include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

These invalidity contentions are based on Defendant's present understanding of the Asserted Claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

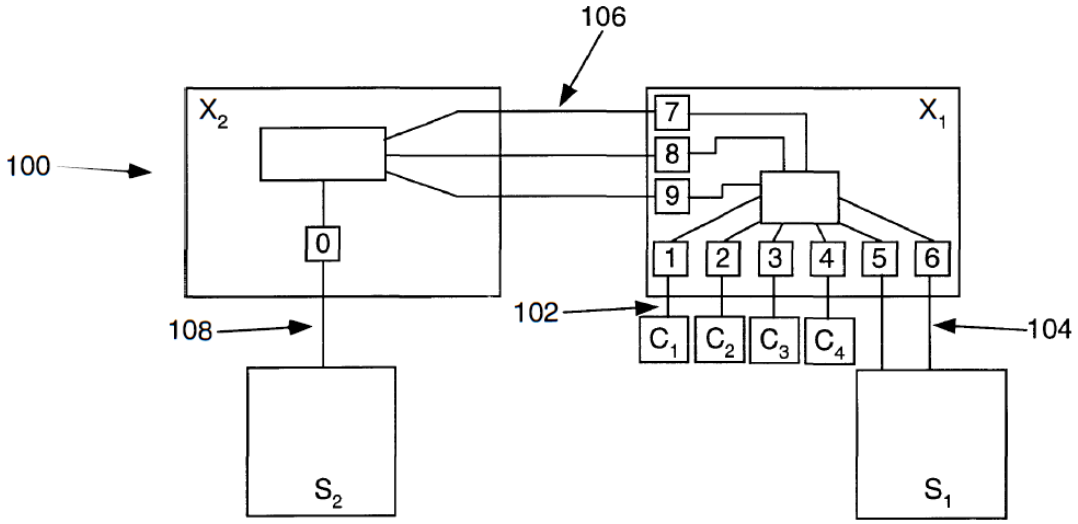
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| 1[preamble] | A method for communication, comprising: | The Reference discloses a method for communication.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 1[a] | coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel, | The Reference discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Abstract ("Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the |

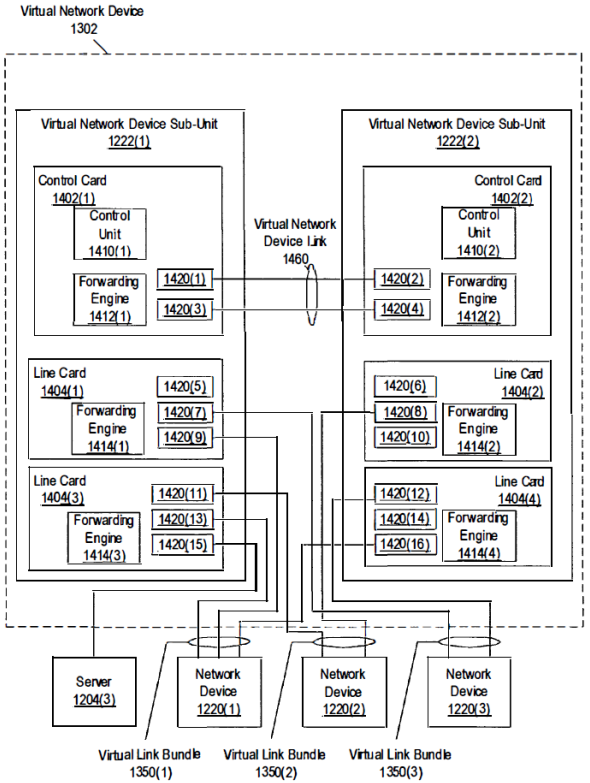
| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | | present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.”)<br><br>DeJager ’424 at 2:24-39 (“While static address distribution improves the efficiency of data transmission over a port group by distributing packet streams among the various ports of a port group, it does not account for the amount of traffic volume of different streams. Accordingly, static address distribution evenly (and thus most efficiently) distributes traffic over the ports of a port group of a switch only if there is the same amount of data being forwarded in each stream. If a given stream is trans-mitting much more than the average amount of data for streams being forwarded through the port group, then there may be inefficiencies in the data transmission in a static address distribution system. For example, this situation may result in there being a long queue for the port to which the heavily loaded stream is assigned, while other ports in the group, which are assigned to more lightly loaded streams, are available to transmit data.”)<br><br>DeJager ’424 at 1:10-28 (“A common computer network implementation includes a plurality of clients, such as personal computers or work stations, connected to each other and one or more servers via a switch or router by network cable. In the present application, the term "switch" is intended to mean any network device that forwards packets from a source to a destination, rather than broadcasting them (i.e., includes router, but excludes repeater). The network is configured to operate at one or more data transmission rates, typically 10 Mbit/sec (e.g., lOBase-T Ethernet), or 100 Mbit/sec (e.g., lO0Base-T Fast Ethernet). More recently, Gigabit data trans-mission rates have become attainable. Data is forwarded on the network in packets which are typically received by a switch from a source network device and then |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | directed to the appropriate destination device. The receipt and transmission of data packets by a switch occurs via ports on the switch. Packets travelling from the same source to the same desti-nation are defined as members of the same stream.")<br><br>DeJager '424 at 2:47-65 ("The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.")<br><br>DeJager '424 at 3:16-39 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of |

| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | | queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.”)<br><br>DeJager ’424 at 9:17-26 (“FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.”)<br><br>DeJager ’424 at 9:43-55 (“At intervals of time (load balancing time intervals) the time mark registers 408 and 410, respectively, are consulted to determine whether or not a stream may change queues (dynamic load balancing), for example as described previously. Each port in the switch's port group 416 preferably has a pair of queue mark and queue mark indicate registers associated with it in order to assist in determining that a port's queue is current and to determine whether the alter-nate time mark register should be cleared and the time mark registers switched, for example as described above. Each packet exits the switch through its assigned port in the port group 416 along one of outbound links 422 to its corresponding destination.”)<br><br>DeJager ’424 at Figure 1 |

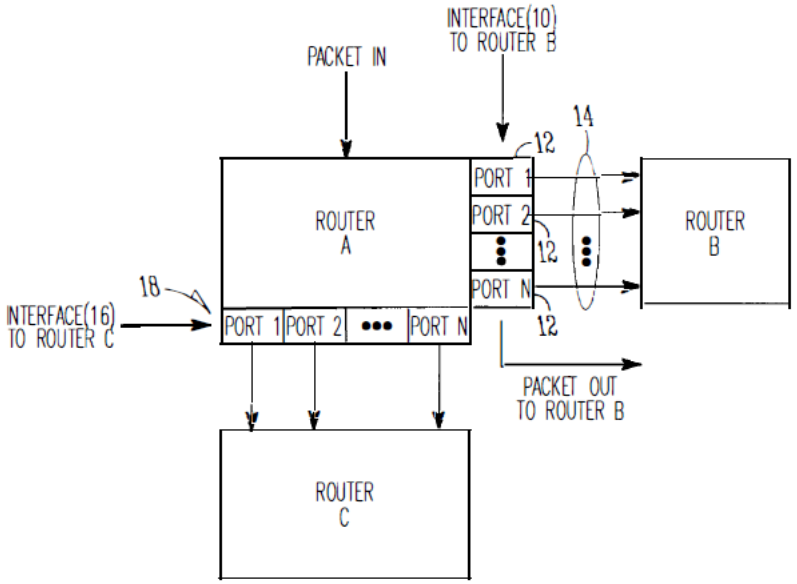
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>***FIG. 1***<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0034] ("Each network device component includes an inter-face (it is noted that each network device component can include several other interfaces as well). Network device component 110(1) includes interface 120(1), network device component 110(2) includes interface 120(2), and network device component 110(3) includes interface 120(3). Inter-faces 120(1 )-120(3) are interfaces of network device 100(1 ). Network device component 110( 4) includes interface 120( 4), network device component 110(5) includes interface 120(5), and network device component 110(6) includes interface 120( 6). Interfaces 120( 4)-120( 6) are interfaces of network device 100(2). Each interface 120(1)-120(6) can be a physical interface or logical interface.")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).")<br><br>Dontu at [0104] ("In virtual network device sub-unit 1222(1), line card 1404(1) includes forwarding engine 1414(1) and inter-faces 1420(5), 1420(7), and 1420(9). Interface 1420(7) is coupled to network device 1220(3). Interface 1420(9) is also coupled to network device 1220(1). Interface 1420(5) is unused in this example. Line card 1404(3) includes forward-ing engine 1414(3), interfaces 1420(11) and 1420(13), and port 1420(15). Interfaces 1420(11) and 1420(13) are respec-tively coupled to network devices 1220(2) and 1220(1). Interface 1420(15) is coupled to server 1204(3). In embodi-ments in which network devices 1220(1)-1220(3) are adjunct network devices controlled by virtual network device 1302, interfaces 1420(7), 1420(9), 1420(11), and 1420(13) are operated as uplink interfaces, while interface 1420(15), which is not coupled to an adjunct network device, is operated as a normal port.")<br><br>Dontu at [0105] ("In virtual network device sub-unit 1222(2), line card 1404(2) includes forwarding engine 1414(2) and inter-faces 1420(6), 1420(8), and 1420(10). Interface 1420(8) |

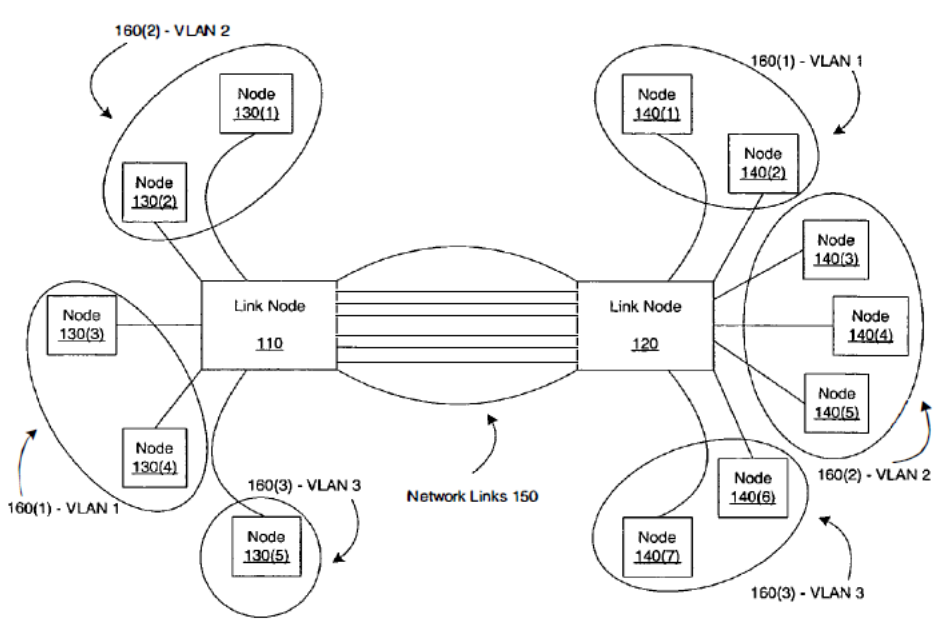
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | is coupled to adjunct network device 1220(2), and interfaces 1420(6) and 1420(10) are unconnected. Line card 1404(4) includes forwarding engine 1414( 4) and interfaces 1420(12), 1420(14), and 1420(16). Interfaces 1420(12) and 1420(16) are respectively coupled to adjunct network devices 1220(3) and 1220(1). Interface 1420(14) is unused. In embodiments in which network devices 1220(1)-1220(3) are adjunct network devices controlled by virtual network device 1302, interfaces 1420(8), 1420(12), and 1420(16) are operated as uplink interfaces,")<br><br>Dontu at Figure 14<br>  <br><br>FIG. 14 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Li '914 at Abstract ("A method and system for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. In one embodiment, a list is created of output ports that are coupled with the adjacent router, and the list is modified based on network traffic. A port is selected from the list of ports, and the packet is transmitted over the selected port. In one example, the list is modified continuously as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, a router can adaptively and evenly distribute the packet trans-mission traffic over the output ports.")<br><br>Li '914 at 2:6-21 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances 20 of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.")<br><br>Li '914 at 2:44-55 ("According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 2:62-3:8 ("According to another embodiment, disclosed herein is a router having at least a first and second output port. The router includes at least one queue for the first output port, the at least one queue for storing packets to be transmitted along the first output port; at least one queue for the second output port, the at least one queue for storing packets to be transmitted along the second output port; a rate counter for measuring transmis-sion rates along the first and second output ports; and means for determining upon which output port a packet should be transmitted. The means for determining may include means for creating a list of output ports that are coupled with the adjacent router, the list including the first and second output ports; and means for continuously modifying the list based on network traffic.")<br><br>Li '914 at 3:58-4:8 ("According to one broad aspect of the invention, disclosed herein is a system and method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. As will be discussed below, in one embodiment, a list is created of output ports that are coupled with the adjacent router, and the list is modified based on network traffic. A port is selected from the list of ports, and the packet is transmitted over the selected port. In one example, the list is modified continuously as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, by determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, a router can adaptively and evenly distribute the packet transmission traffic over the output ports. Various embodiments of the invention will now be discussed.")<br><br>Li '914 at Figure 1 |

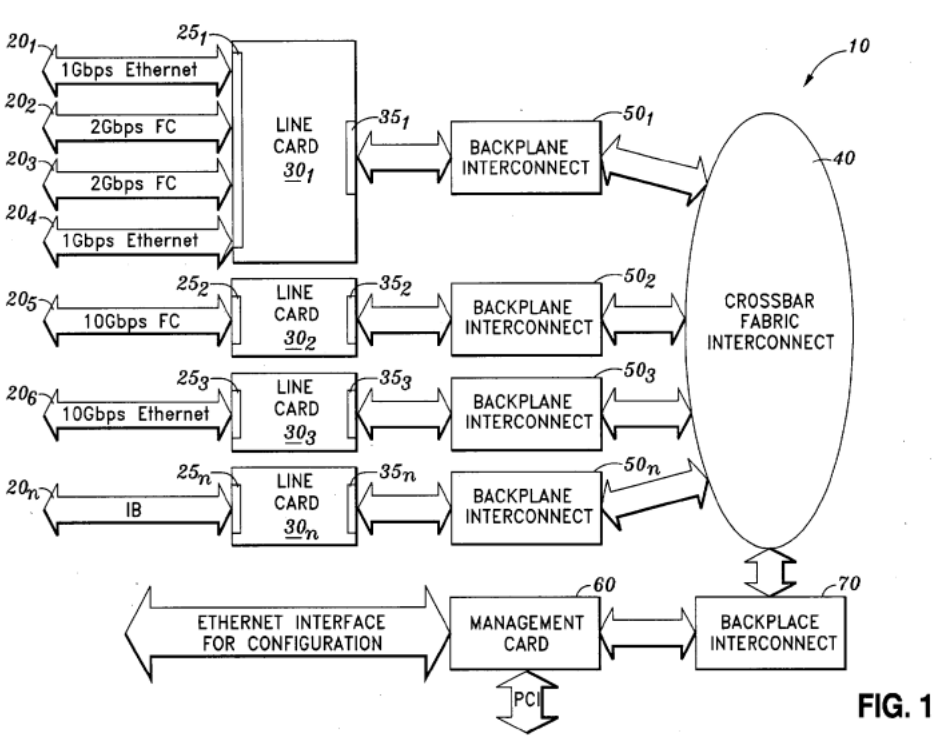
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br><br>Borgione '125 at Abstract ("A method, system, and apparatus to transmit replicated mul-ticast packets over a plurality of physical network links that are combined into one logical channel or link so that the replicated multicast packets are distributed over more than one network link is disclosed. It is further disclosed that distribution over the network links is accomplished, in part, through analyzing the multicast packet for information other than ethernet addresses. Such information can include a tag header including destination interface information.")<br><br>Borgione '125 at 3:60-4:2 ("The present invention presents a method, system, and apparatus to transmit replicated multicast packets over a plu-rality of physical network links that are combined into one logical channel or link so that the replicated multicast packets are |

| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | | distributed over more than one network link. This is accomplished, in part, through analyzing the multicast packet for information other than Ethernet addresses. Such informa-tion can include a tag header including destination interface information (for example, a VLAN identification field in an IEEE Std. 802.lQ packet header tag).”)<br><br>Borgione ’125 at 4:3-9 (“Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.”)<br><br>Borgione ’125 at 2:8-18 (“Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.”)<br><br>Borgione ’125 at 4:14-21 (“Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.”)<br><br>Borgione ’125 at 4:22-30 (“Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select |

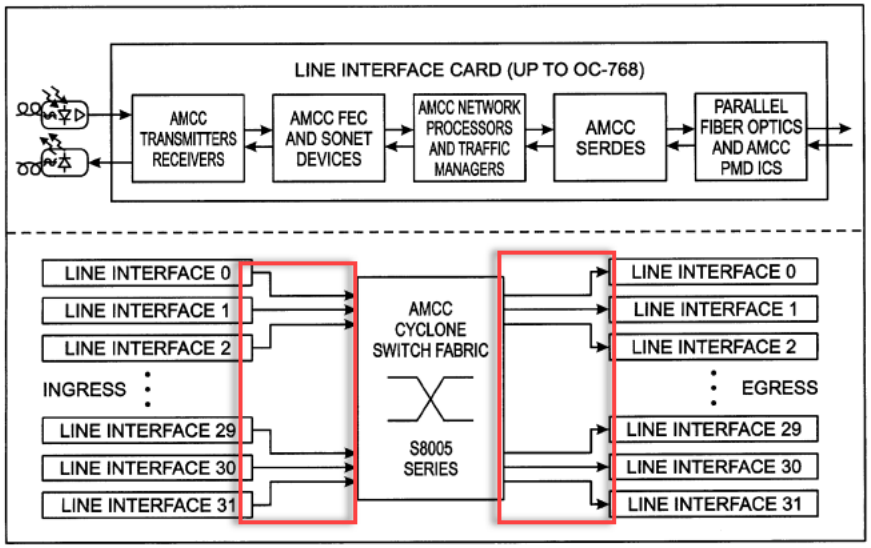
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at Figure 1<br>  <br><br>**Figure 1** |
| 1[b] | at least one of said first physical links being a bi-directional link operative to | The Reference discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. |

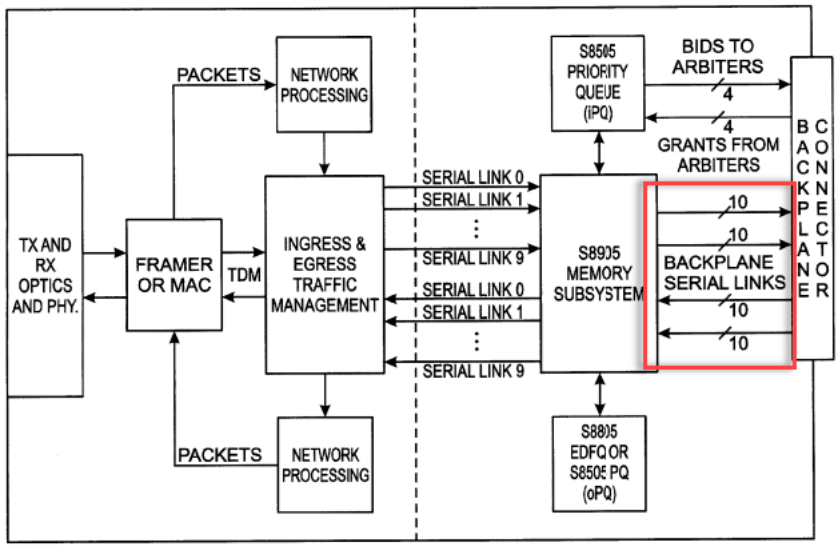
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | communicate in both an upstream direction and a downstream direction | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 1[c] | coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel, | The Reference discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, Dontu, and Smith '427.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable |

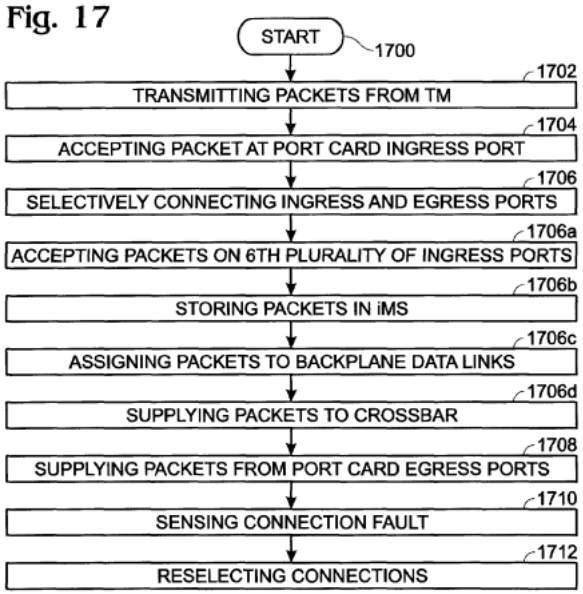
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

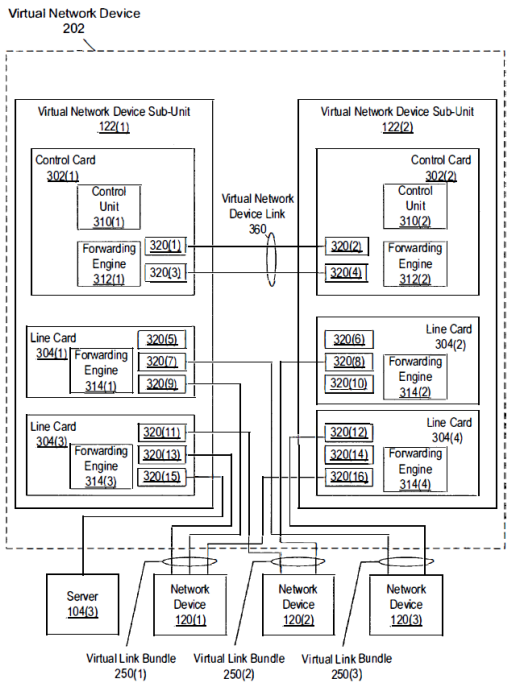
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

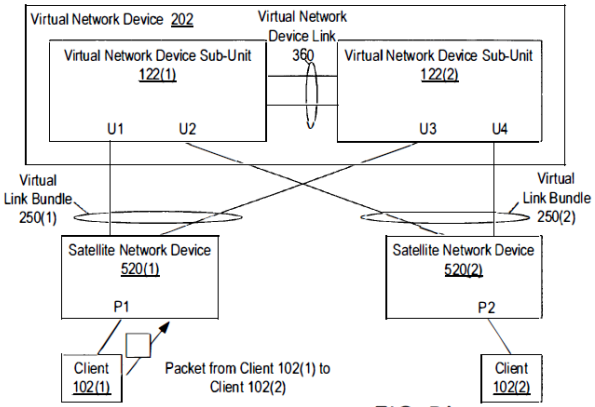
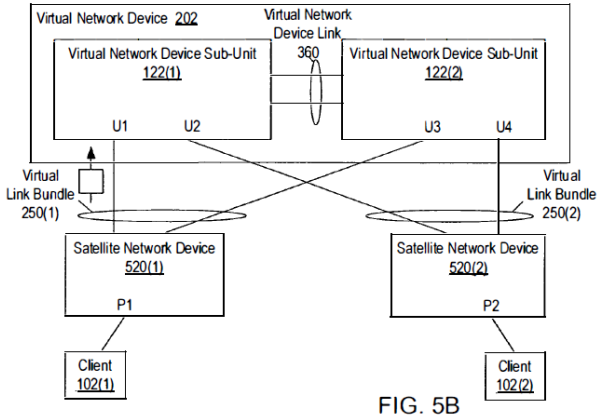
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.")<br><br>Further, Cisco innovated and patented the accused multi-chassis link aggregation features before Orckit, which involves a second group of physical links coupling line cards or interface modules of one apparatus to another.<br>• Smith '430<br>• Smith '427<br><br>Smith '430 at Figure 3 |

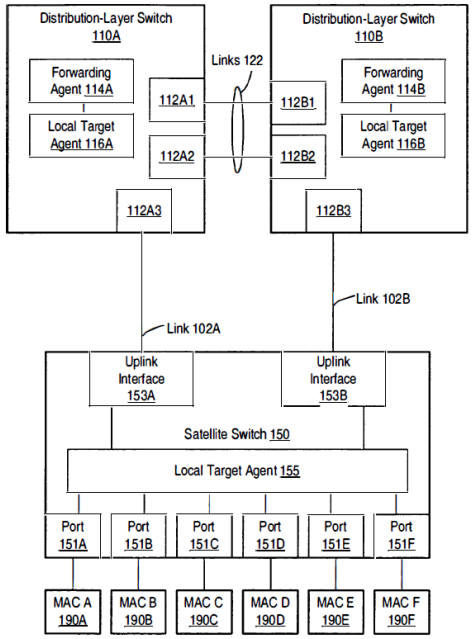
| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br><br>Smith ’430 at Figures 5A and 5B |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 5A<br>  <br>FIG. 5B<br><br>Smith '430 at 2:21-43 ("In such embodiments, the first virtual network device sub-unit can be configured to maintain consistent forwarding information with the second virtual network device sub-unit. For example, in one embodiment, the controller (in the first virtual network device sub-unit) is configured to perform control protocol processing for the first interface according to a routing protocol running on the interface bundle. The con-troller is configured to provide information generated when performing the control protocol processing to a |

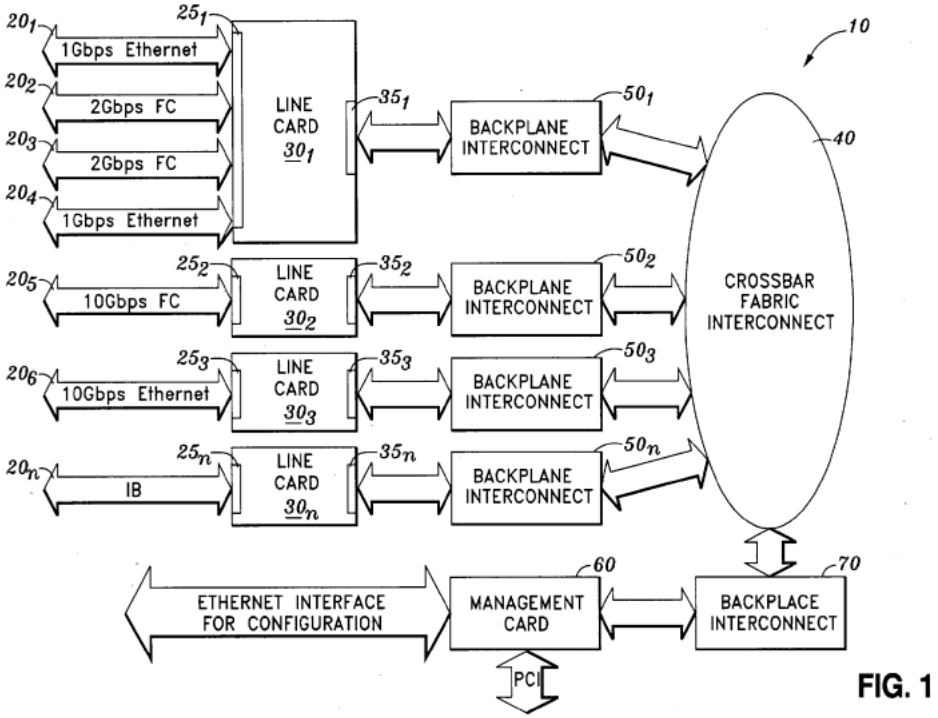
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | secondary controller comprised in the second virtual network device sub-unit. The secondary controller is configured to use the information to manage the second interface.<br>One embodiment of a method involves: assigning a first logical identifier to each interface included within an inter-face bundle, where the interface bundle includes a first inter-face of a first virtual network device sub-unit and a second interface of second virtual network device sub-unit; coupling a first end of a first link to the first interface, the first link included within a virtual link bundle; and coupling a first end of second link to the second interface, the second link also included within the virtual link bundle. The second end of each of the first link and the second link are coupled to a third network device.")<br><br>Smith '430 at 5:10-50 ("Multiple links can be implemented between devices in different network layers to provide additional redundancy. For example, as shown in FIG. 1, each network device 120 (1)-120(n) in access layer 110 can be coupled to distribution layer 112 by two ( or more) different links. Similarly, each network device 122(1 )-122(n) in distribution layer 112 can be coupled to core layer 114 by two ( or more) different links. In one embodiment, each link is an Ethernet link.<br>Within each network layer, multiple redundant network devices can be configured to collectively operate as a single virtual network device. For example, as shown in FIG. 1, two or more network devices in distribution layer 112 can operate as a virtual network device 202. Similarly, two or more of network devices 124(1 )-124(n) can operate as a single virtual network device 204, and two or more of network devices 126(1)-126(n) can operate as a single virtual network device 206. More details of how two distribution-layer network devices can collectively operate as a distribution-layer virtual network device 202 are shown in FIGS. 2A, 2B, and 3. Virtual network devices can be coupled to other virtual network devices, to network devices, and/or to clients and/or servers by virtual link bundles, as described below. In general, any multi-ported device (whether a physical device, such as a network device, client, or server, or a virtual network device) can be coupled to a virtual network device by a virtual link bundle that includes several links, some of which terminate on different sub-units within the virtual network device.<br>FIG. 2A shows an example of a network in which there are two network devices 120(1) and 120(2) in access layer 110. There are also two network devices 122(1) and 122(2) in distribution layer 112. These two network devices 122(1) and 122(2) operate as a single |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | virtual network device 202 in this example. Each network device 120(1)-120(2) is coupled to distribution layer 112 by two links. In this example, each of those two links is coupled to a different one of network devices 122(1) and 122(2). This provides redundancy, allow-ing network devices 120(1) and 120(2) to continue to com-municate with distribution layer 112 even if one of network devices 122(1) or 122(2) fails or if one of the links between a given access-layer network device and a given distribution-layer network device fails.")<br><br>Smith '430 at 6:25-45 ("In embodiments, such as the one shown in FIG. 2B, in which network devices 120(1) and 120(2) see themselves as being connected to a single network device, the use of a virtual link bundle is simplified. For example, if network device 120(1) is aware that virtual link bundle 250(1) termi-nates at two different network devices, network device 120(1) can select a link on which to send a particular packet based on Spanning Tree Protocol. The use of Sparming Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 120(1) simply views virtual network device 202 as a single entity. When viewing virtual network device 202 as a single entity, for example, network device 120(1) can simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 250(1) fails, there is no need for network device 120(1) to change how Spanning Tree Protocol is applied. Instead, network device 120(1) can simply continue to use the non-failed links within virtual link bundle 250(1).")<br><br>Smith '427 at Abstract ("A method may involve: receiving a packet ( e.g., via a port or uplink interface in a satellite switch) and conveying the packet between one or more ports and one of several uplink interfaces. The one or more ports and the uplink interface are associated with each other. The association can be indepen-dent ofVLAN (Virtual Local Area Network). As an example, such a method can involve: receiving a first packet via a first port; conveying the first packet to a distribution-layer via a first uplink interface; receiving a second packet via a second port; and conveying the second packet to the distribution-layer via a second uplink interface, where the first uplink interface is associated with the first port |

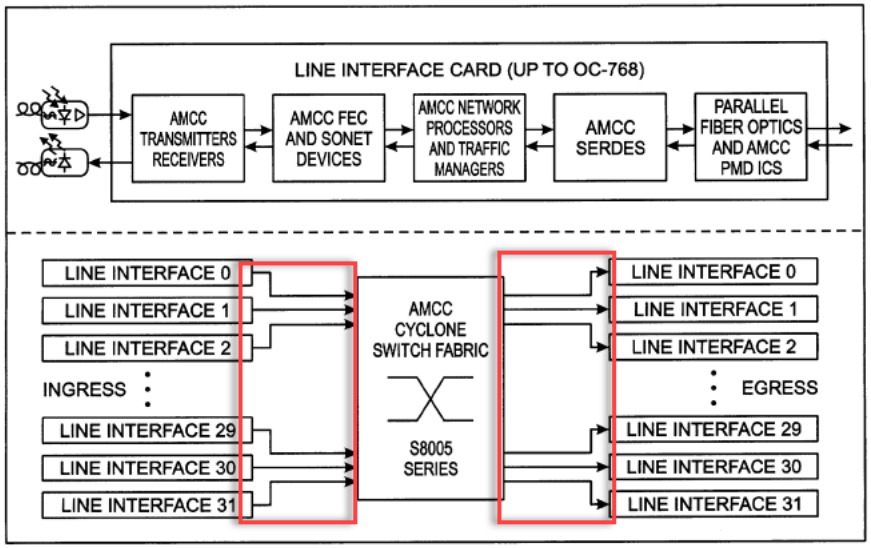
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | and the second uplink interface is associated with the second port. In some embodi-ments, ports and uplink interfaces are associated by being assigned to the same virtual linecard.")<br><br>Smith '427 at 2:28-55 ("A method may involve: receiving a packet ( e.g., via a port or uplink interface in a satellite switch) and conveying the packet between one or more ports and one of several uplink interfaces. The one or more ports and the uplink interface are associated with each other. The association can be indepen-dent of VLAN (Virtual Local Area Network). As an example, in one embodiment, such a method can involve: receiving a first packet via a first port; conveying the first packet to the distribution-layer via a first uplink interface; receiving a sec-ond packet via a second port; and conveying the second packet to the distribution-layer via a second uplink interface, where the first uplink interface is associated with the first port and the second uplink interface is associated with the second port. The first port can be associated with the same VLAN as the second port. By communicating packets between only associated ports and uplink interfaces, undesirable bridging loops can be avoided. Several of the ports can be associated with the same uplink interface, and several uplink interfaces can be associated with the same port.<br>In some embodiments, a system includes several ports, several uplink interfaces, and a local target agent configured to convey packets between the ports and uplink interfaces. The local target agent is configured to convey a packet between one of the ports and one of the uplink interfaces. The one of the ports and the one of the uplink interfaces are associated with each other ( e.g., by being assigned to the same virtual linecard). Ports associated with different VLAN s can be assigned to the same virtual linecard.")<br><br>Smith '427 at Figure 2 |

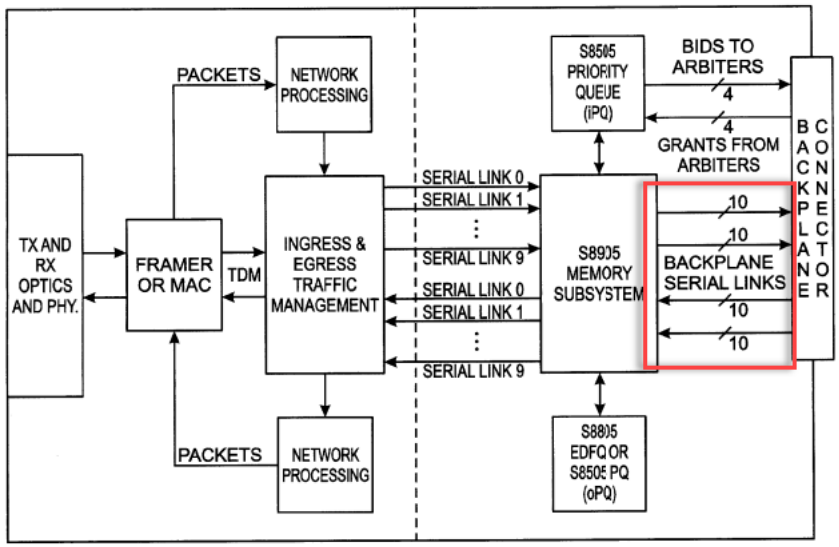
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 2 |
| 1[d] | at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction; | The Reference discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, and Singh. |

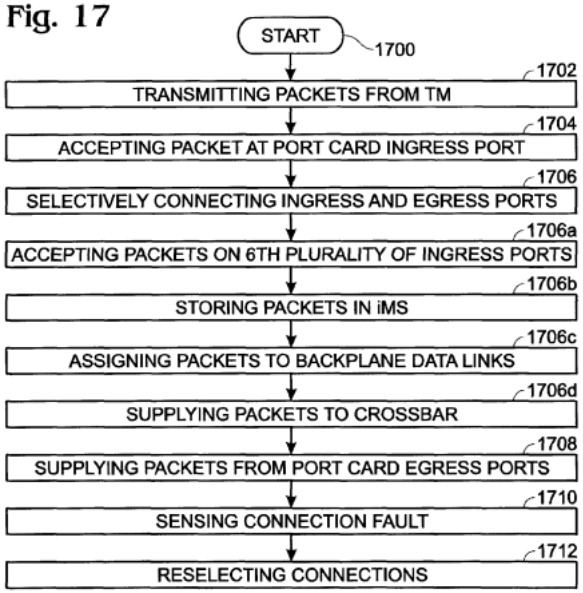
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

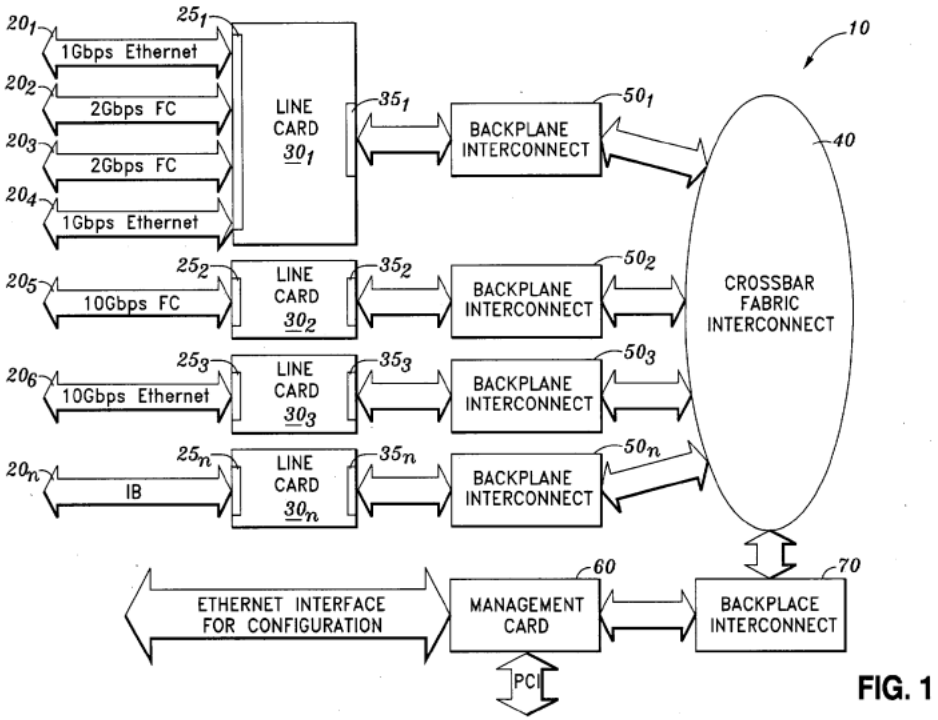
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

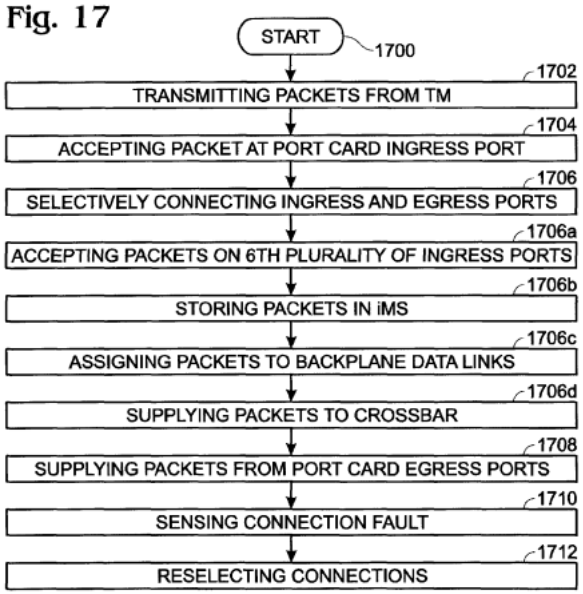
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br> |
| 1[e] | receiving a data frame having frame attributes sent between the communication network and the network node: | The Reference discloses receiving a data frame having frame attributes sent between the communication network and the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

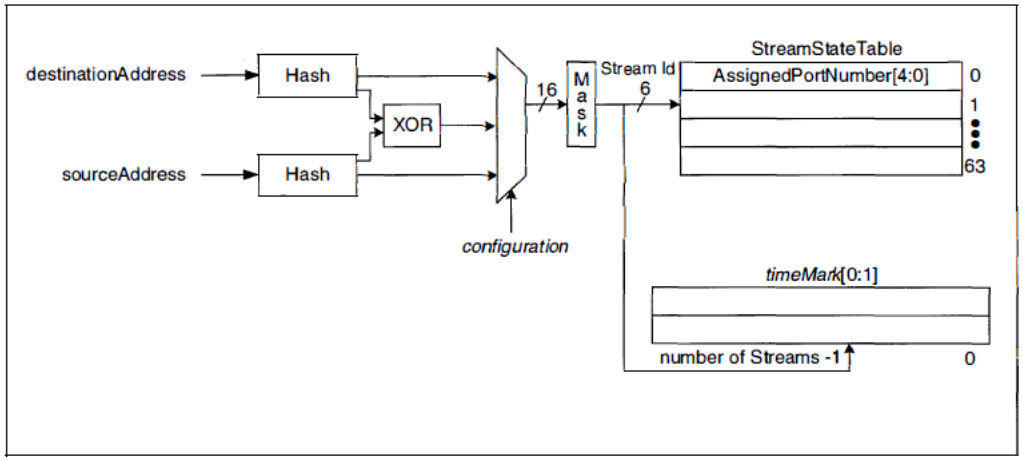
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| 1[f]: | selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and | The Reference discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more |

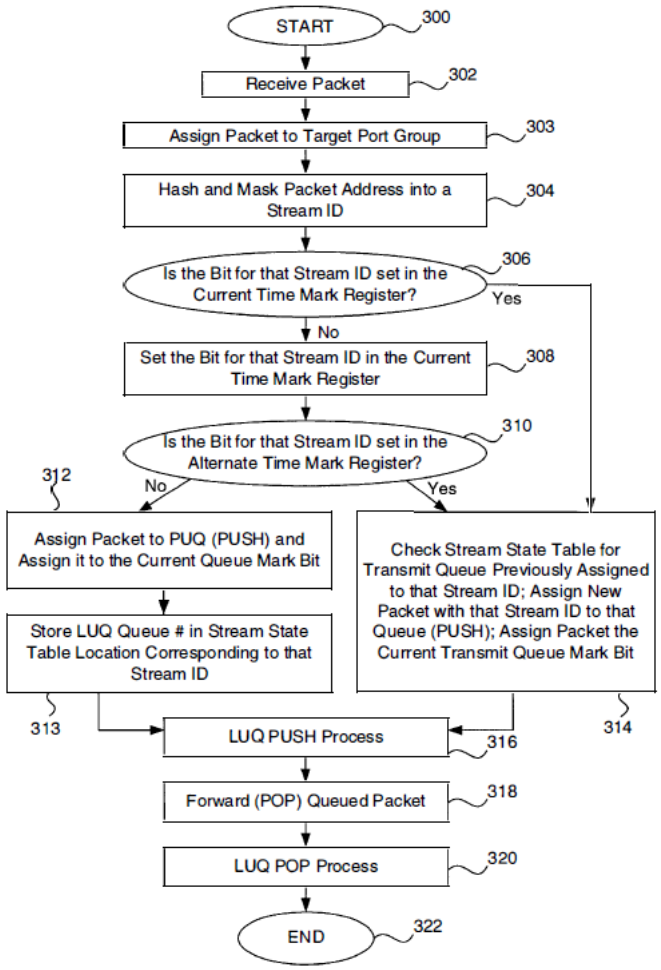
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1<br>  <br><br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including selecting physical links over which to send a packet. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | • Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and |

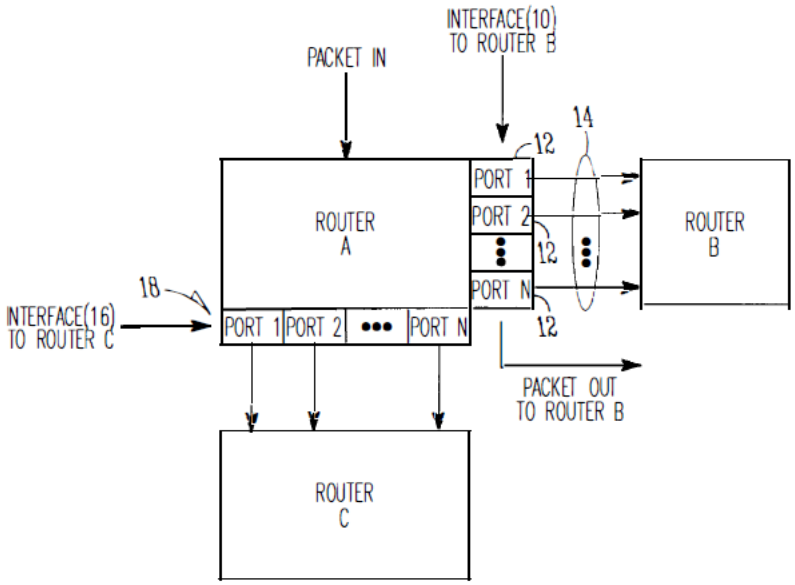
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

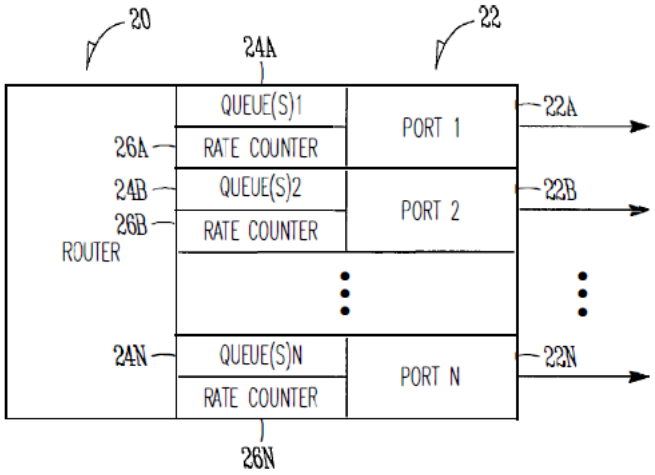
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

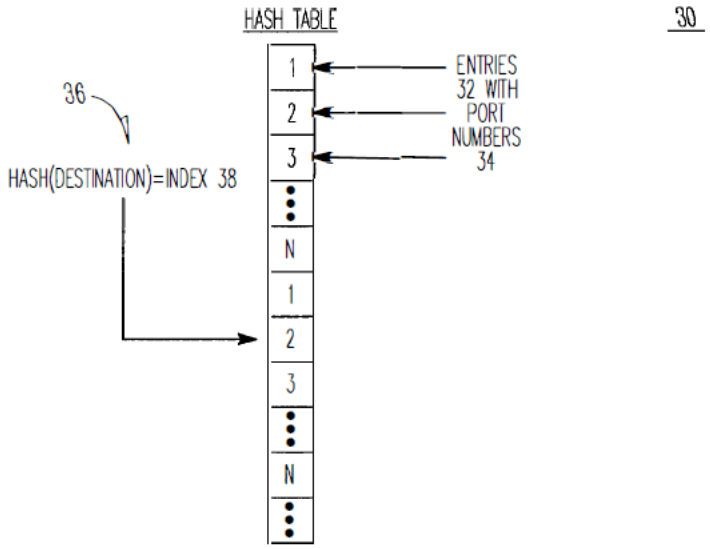
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub- |

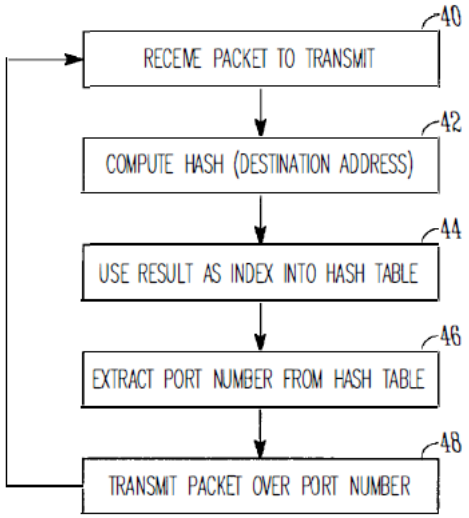
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.") |

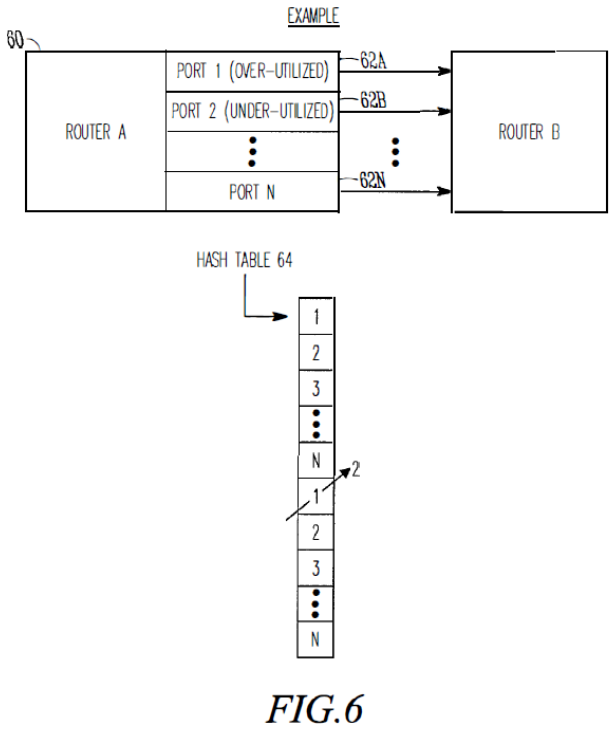
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

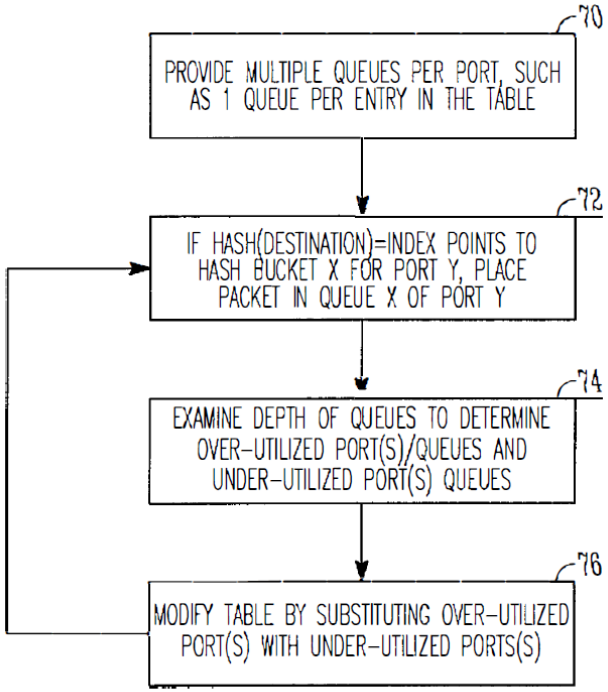
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

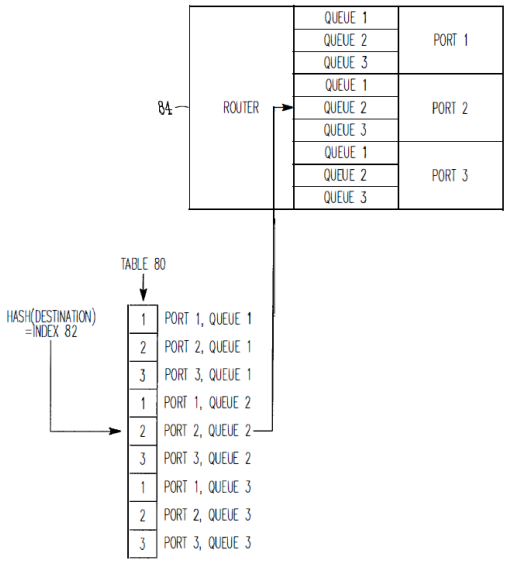
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router |

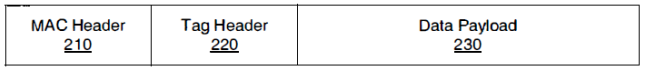
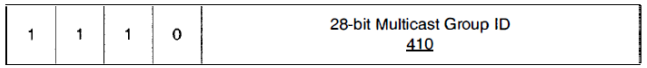
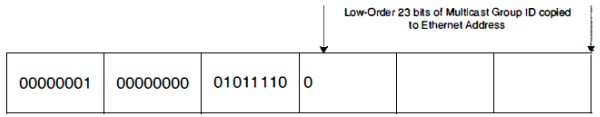
| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | | A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

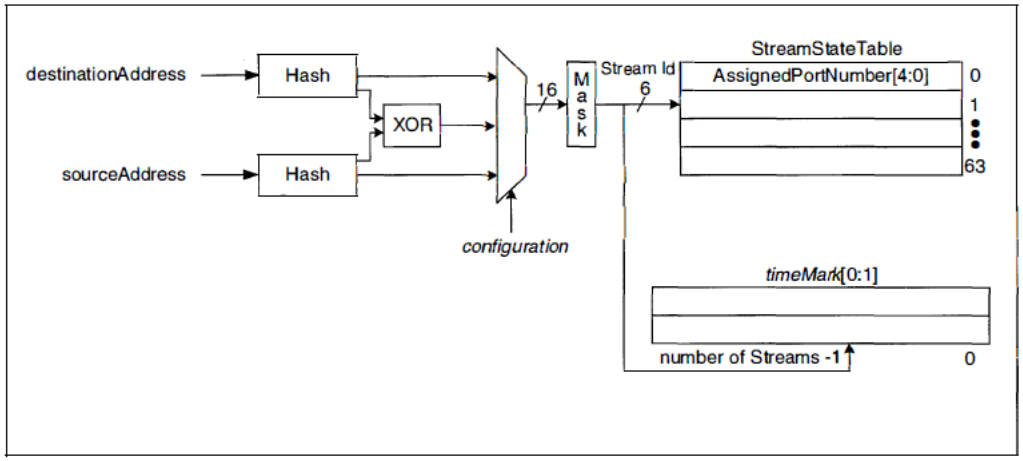
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.") |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).") |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.")<br><br>Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |
| 1[g] | sending the data frame over the selected first and second physical links, | The Reference discloses sending the data frame over the selected first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|

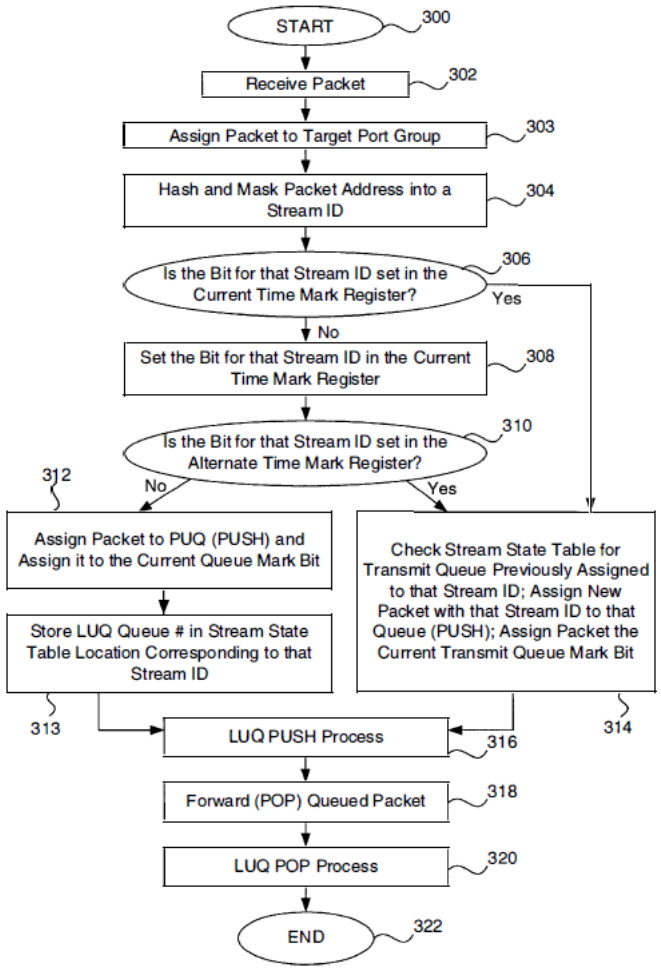



***FIG. 3A***

DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and

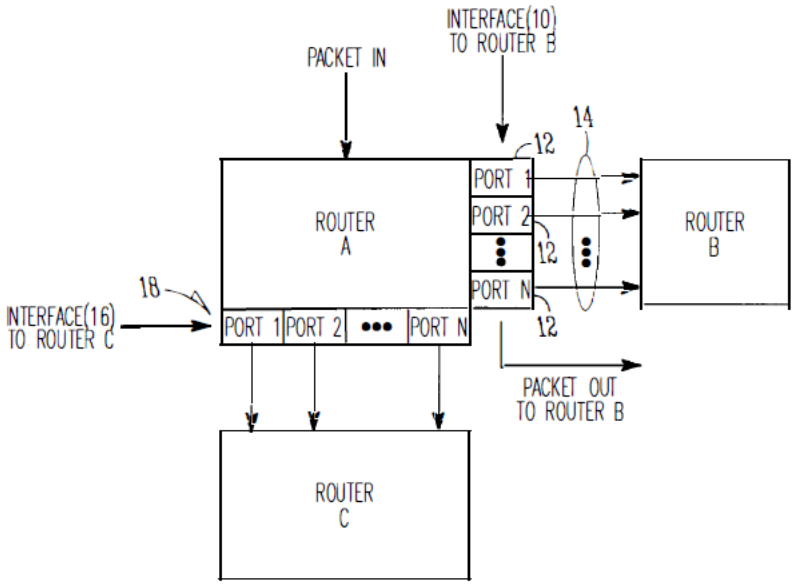
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

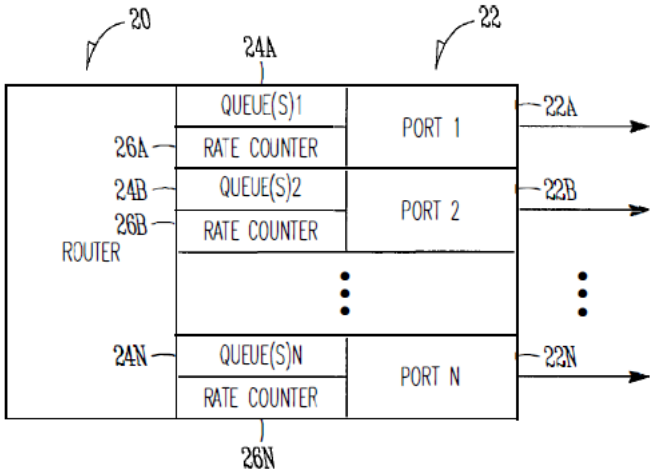
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

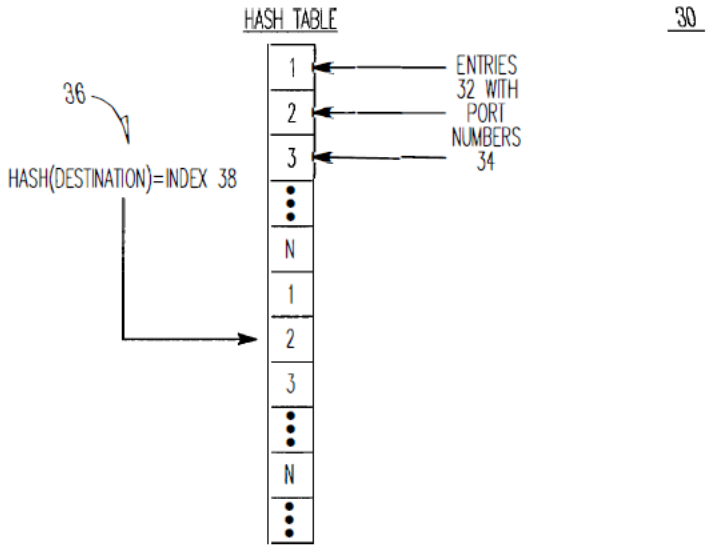
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub- |

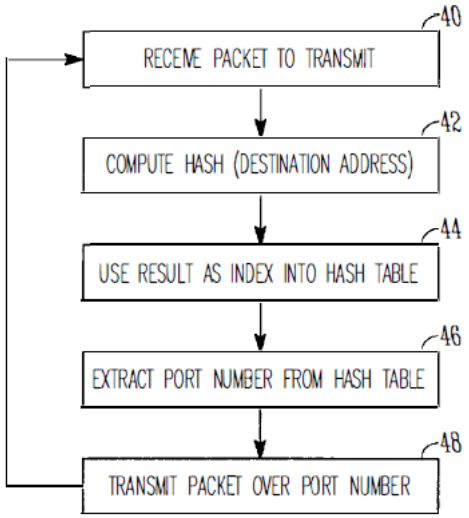
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.") |

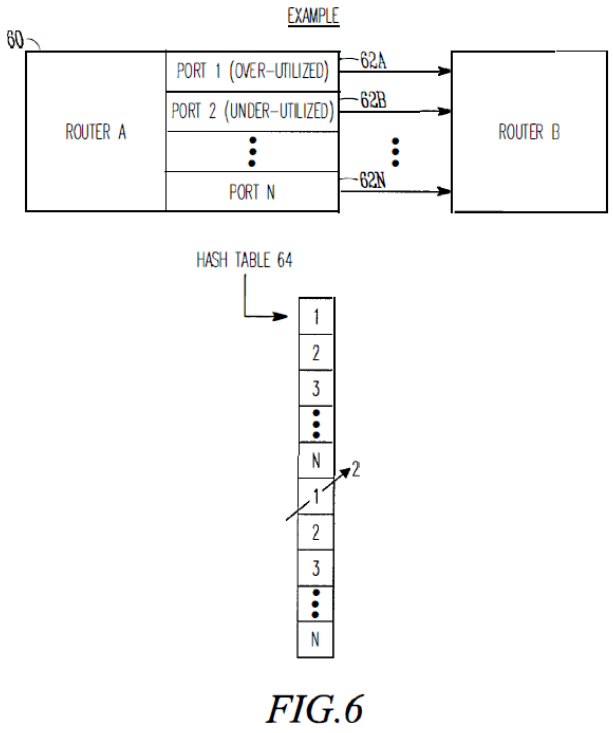
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

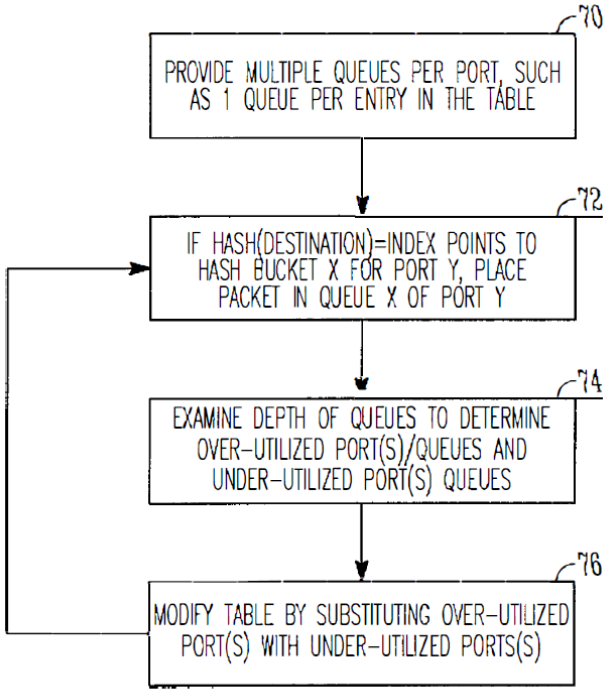
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

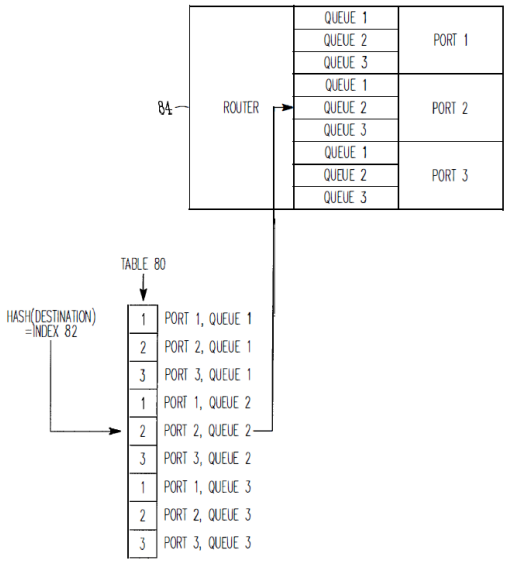
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router |

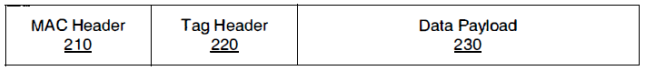
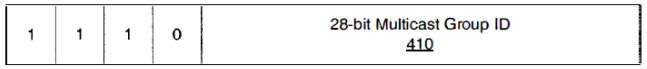
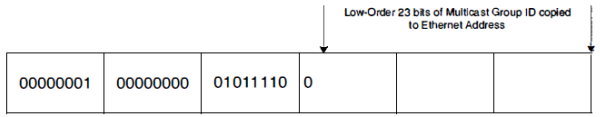
| No. | ’740 Patent Claim 1 | The Reference |
|---|---|---|
| | | A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.") |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).") |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.")<br><br>Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5 |

| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |

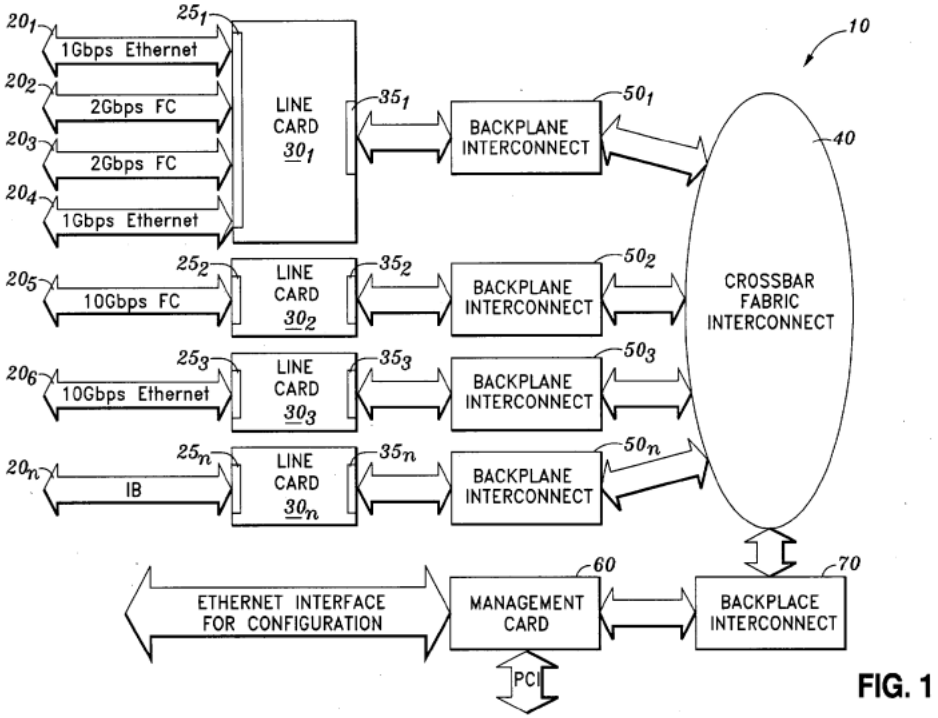
| No. | '740 Patent Claim 1 | The Reference |
|---|---|---|
| 1[h] | said sending comprising communicating along at least one of said bi-directional links. | The Reference discloses said sending comprising communicating along at least one of said bi-directional links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

| No. | '740 Patent Claim 2 | The Reference |
|---|---|---|
| 2[a] | The method according to claim 1, wherein the network node comprises a user node, and | The Reference discloses the method according to claim 1, wherein the network node comprises a user node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 2[b] | wherein sending the data frame comprises establishing a communication service between the user node and the | The Reference discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

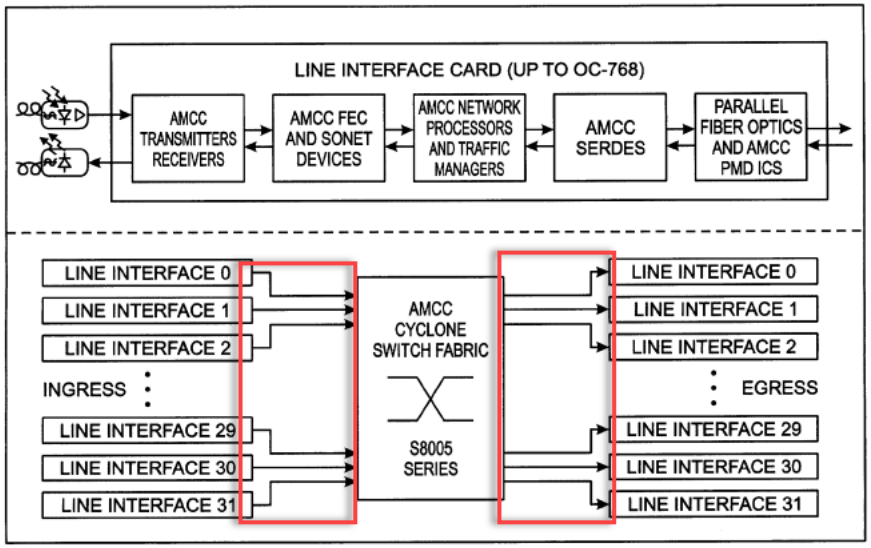
| No. | '740 Patent Claim 2 | The Reference |
|---|---|---|
| | communication network. | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

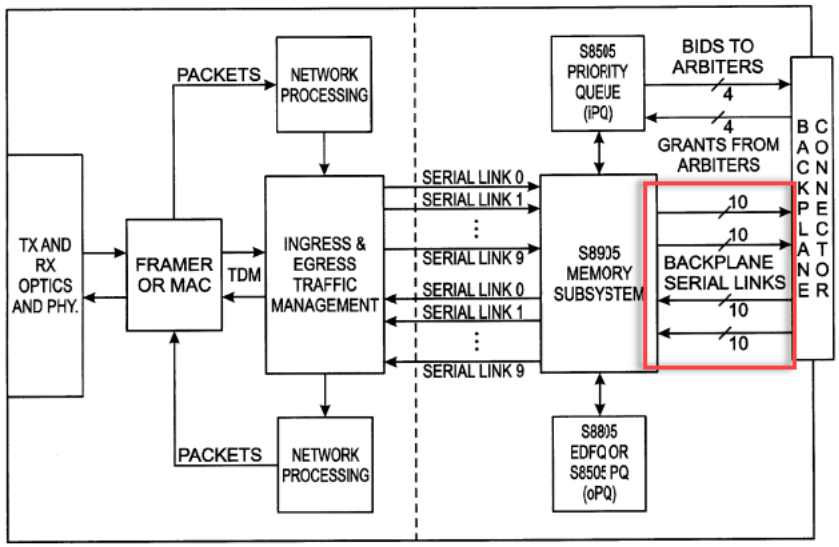
| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| 3 | The method according to claim 1, wherein the second physical links comprise backplane traces formed on a back plane to which the one or more interface modules are coupled. | The Reference discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a back plane to which the one or more interface modules are coupled.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be |

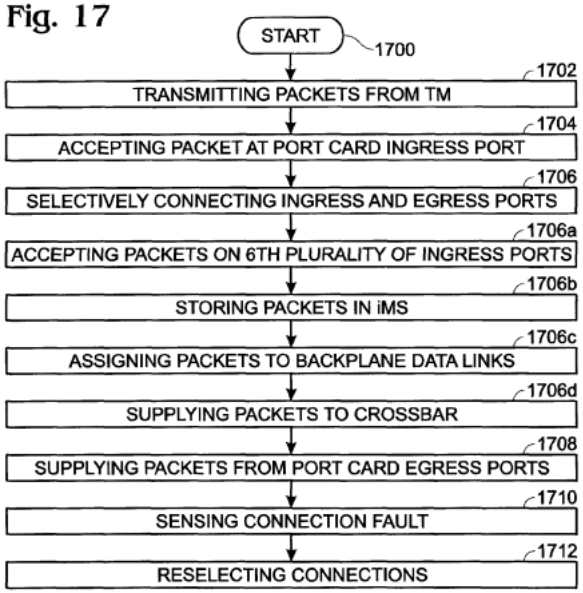
| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| | | appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

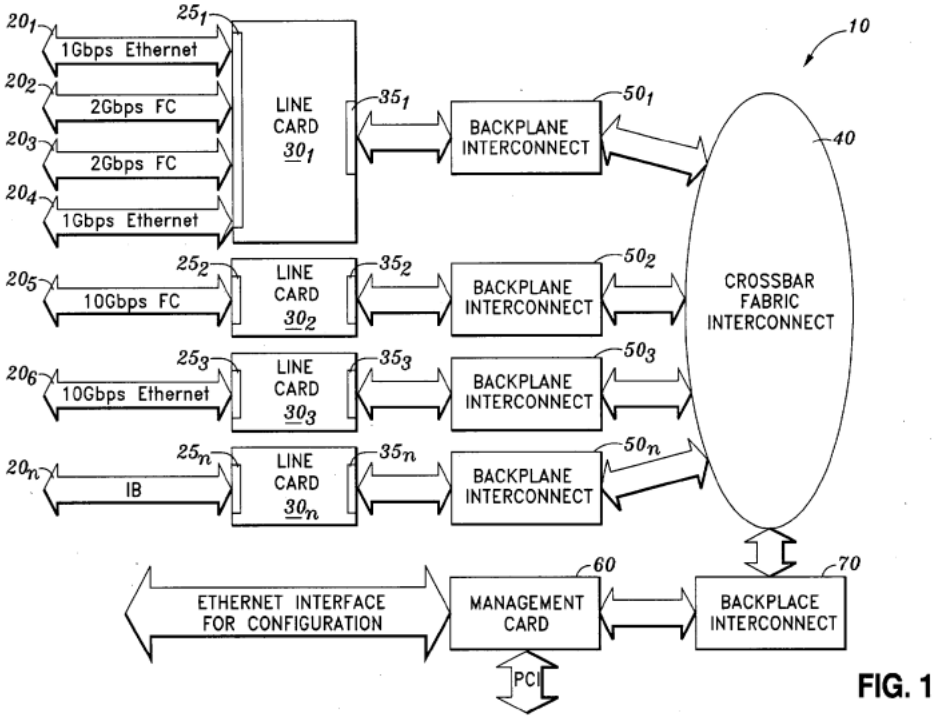
| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 3 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |

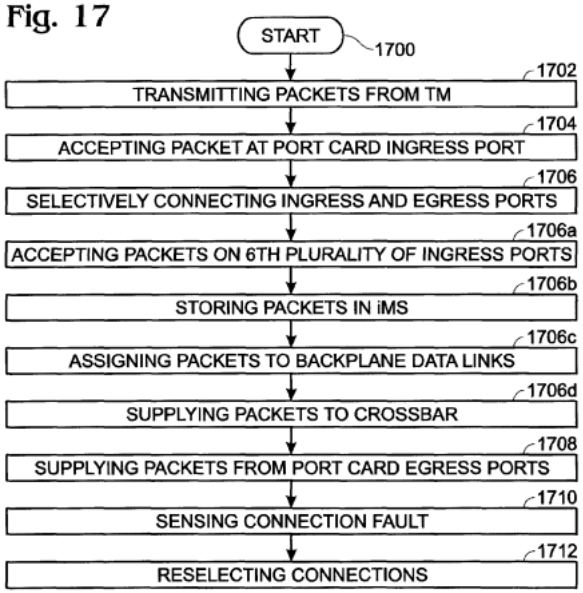
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| 4[preamble] | A method for communication, comprising: | The Reference discloses a method for communication.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 4[a] | coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel; | The Reference discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 4[b] | coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel; | The Reference discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 4[c] | receiving a data frame having frame attributes sent between the communication network and the network node: | The Reference discloses receiving a data frame having frame attributes sent between the communication network and the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 4[d] | selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and | The Reference discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified |

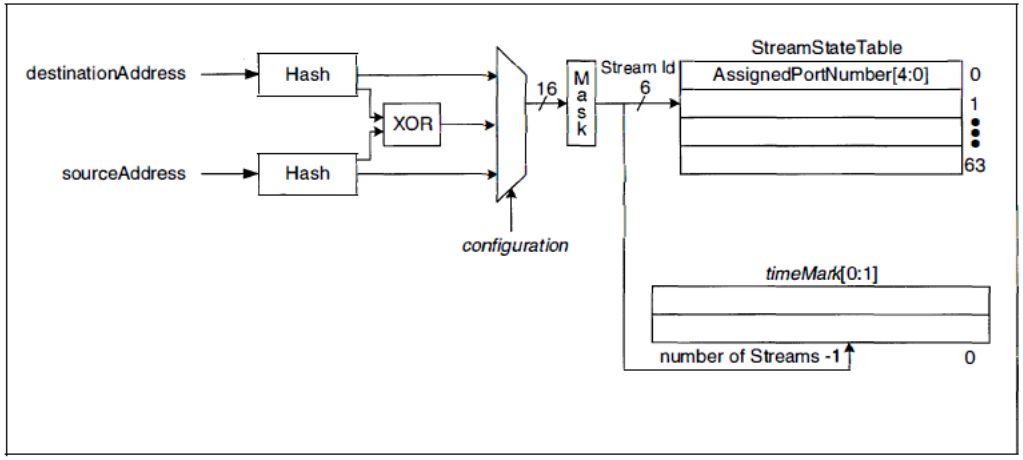
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

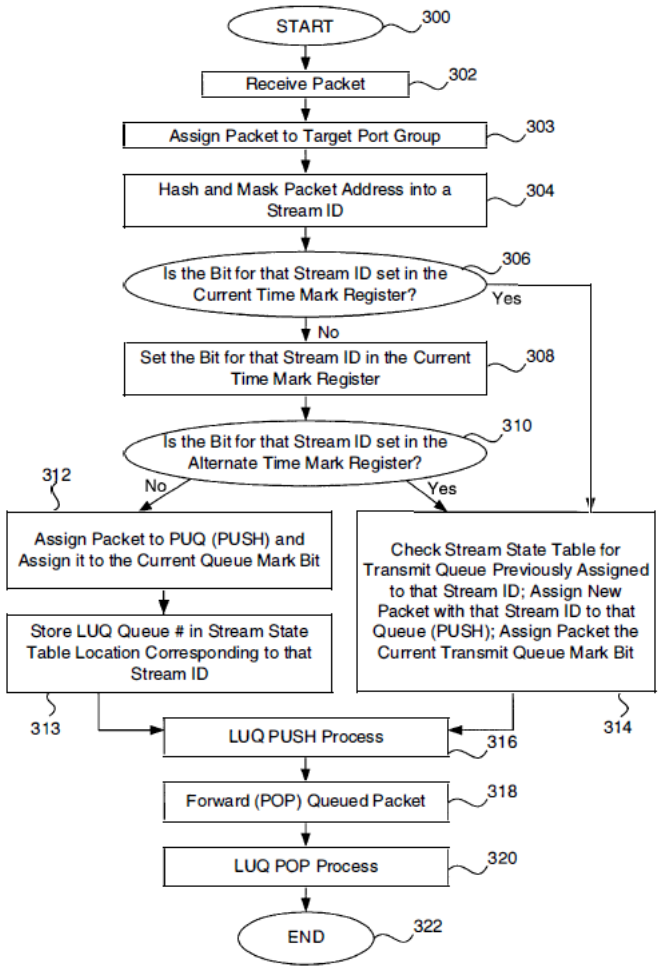
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including selecting physical links over which to send a packet. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and |

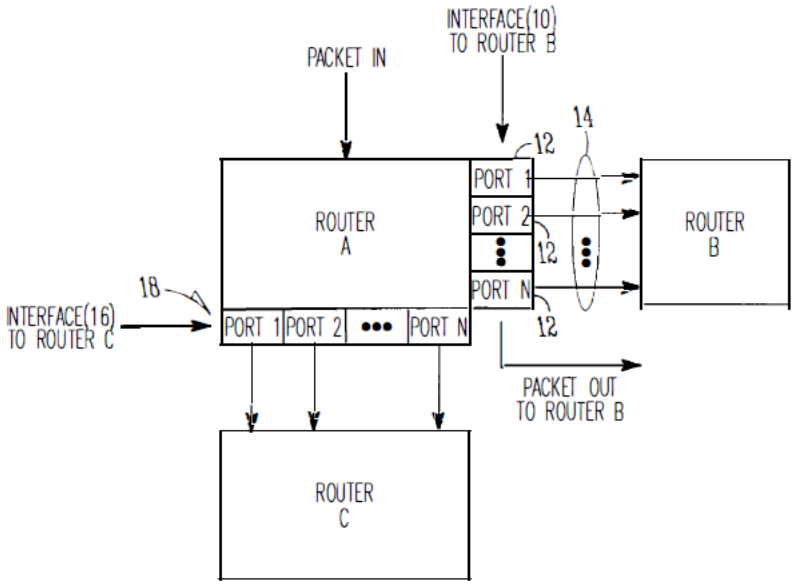
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

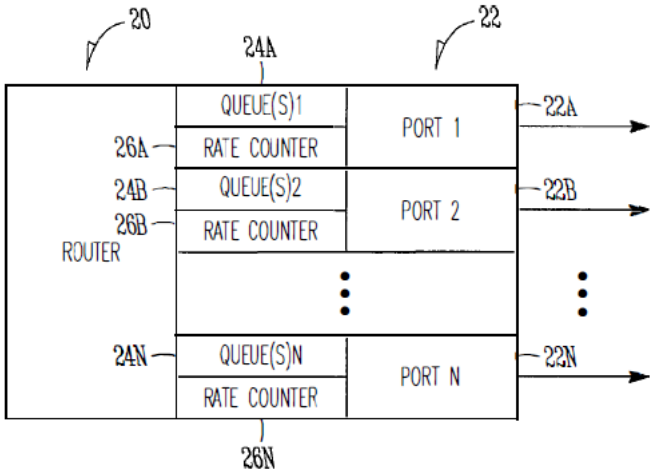
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

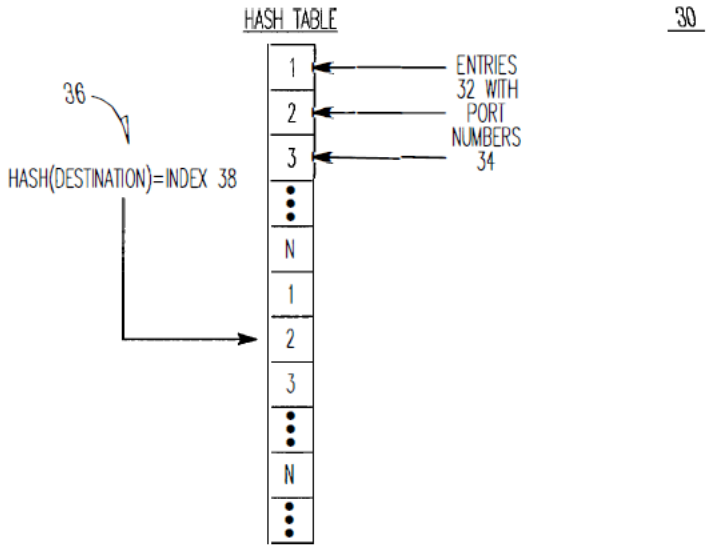
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet |

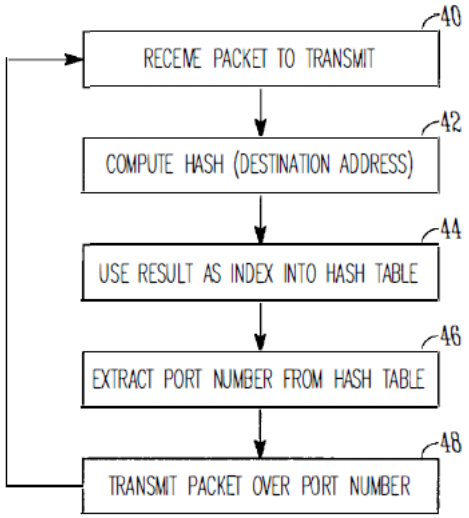
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and |

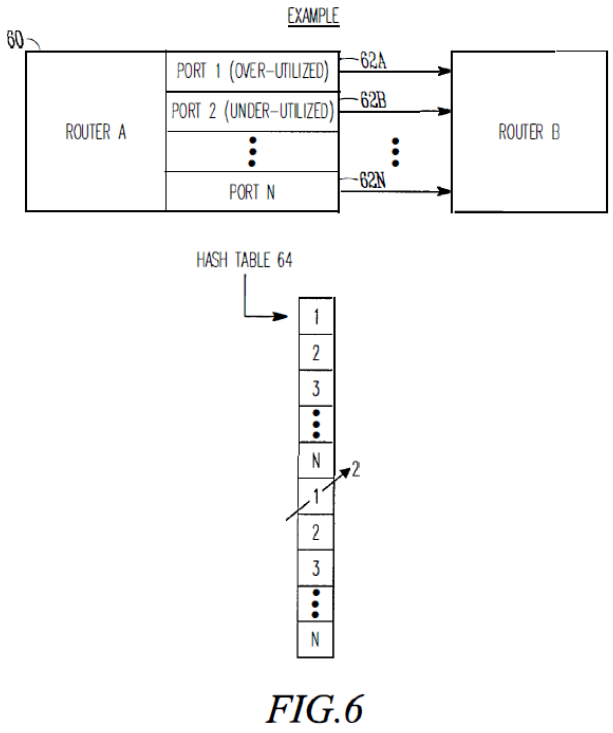
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

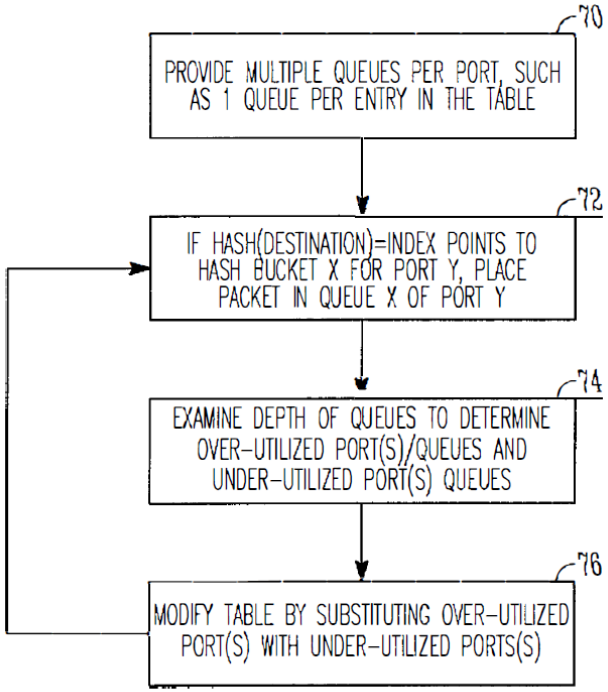
| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li ’914 at Figure 2 |

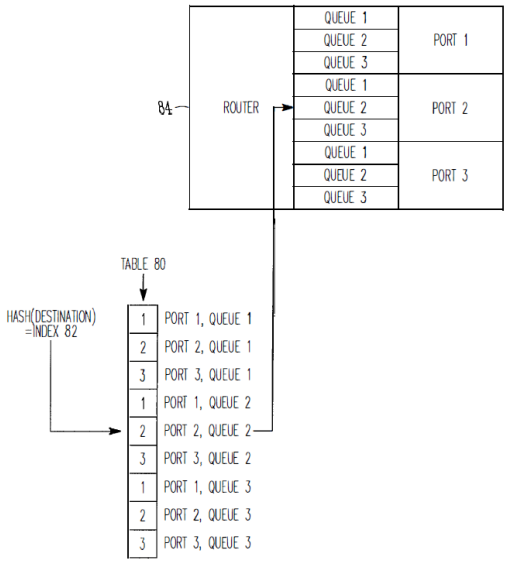
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of |

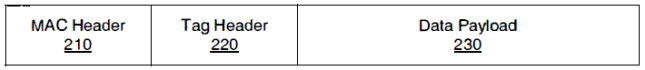
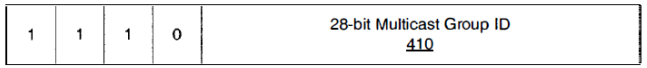
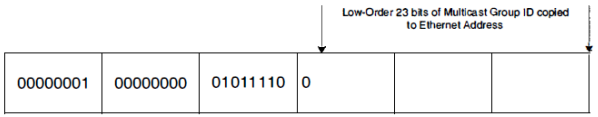
| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

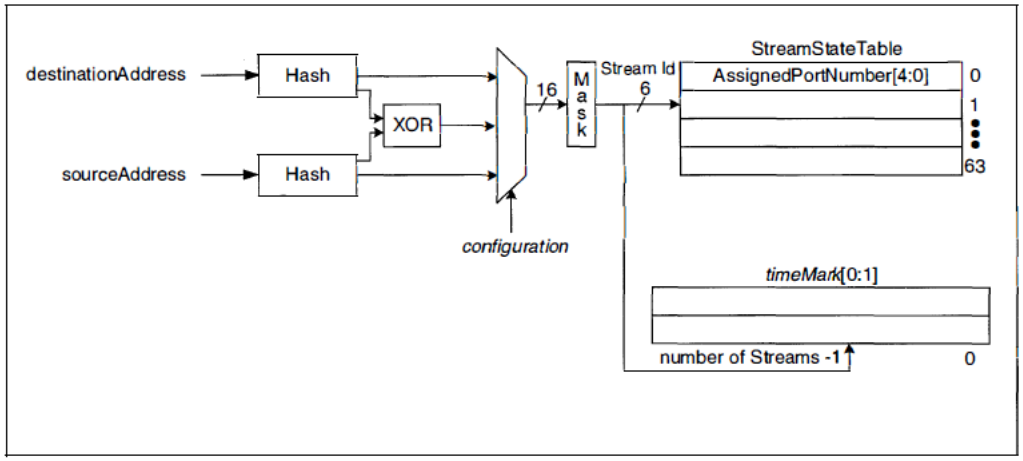
| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.”)<br><br>Li ’914 at 5:13-28 (“In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.”)<br><br>Li ’914 at 5:29-41 (“Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.”) |

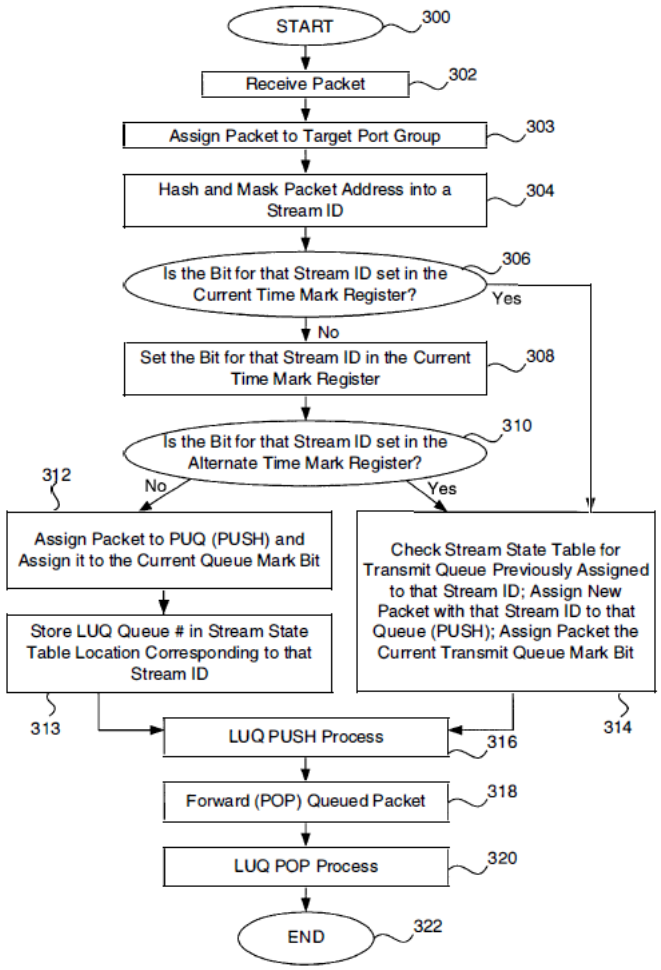
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).") |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.”)<br><br>Borgione ’125 at 5:28-54 (“The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.”)<br><br>Borgione ’125 at 7:1-5 (“A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).”)<br><br>Borgione ’125 at Figure 2-5 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |
| 4[e] | sending the data frame over the selected first and second physical links, | The Reference discloses sending the data frame over the selected first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and |

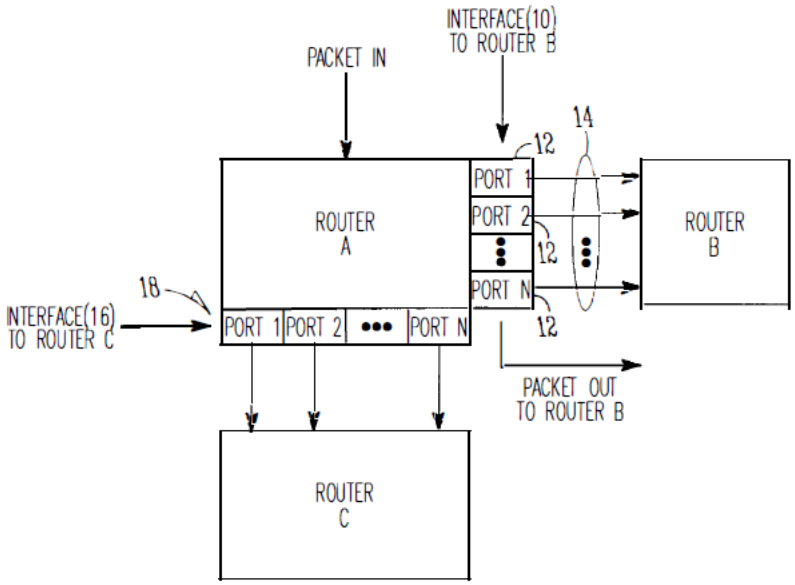
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

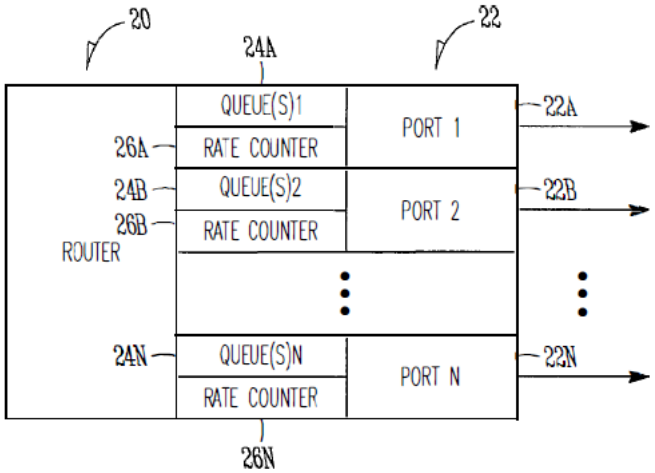
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

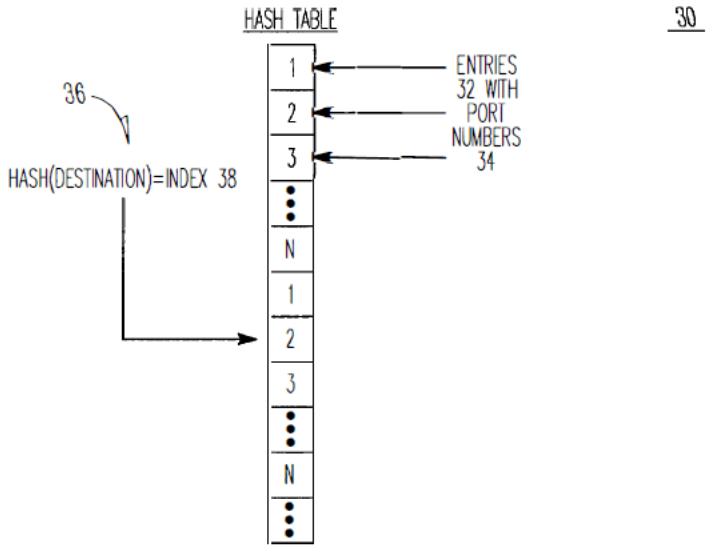
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet |

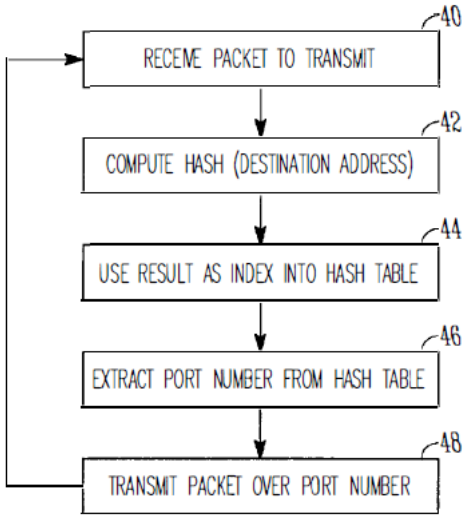
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and |

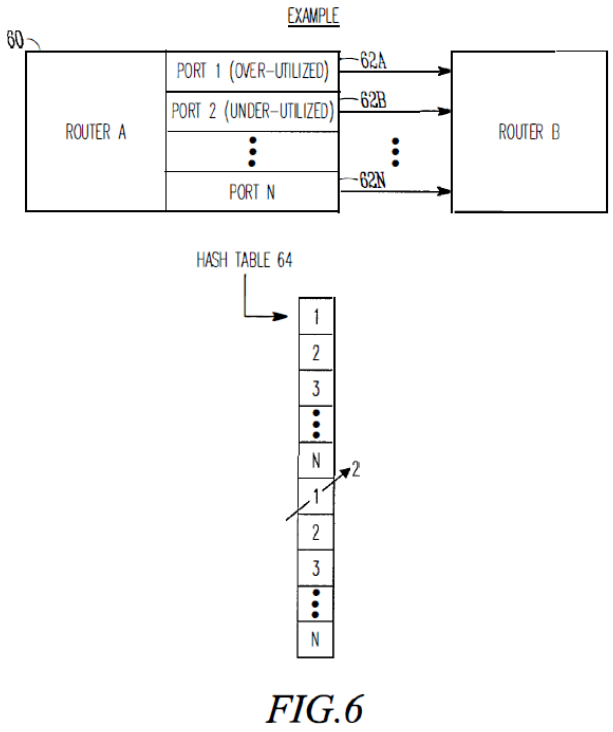
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

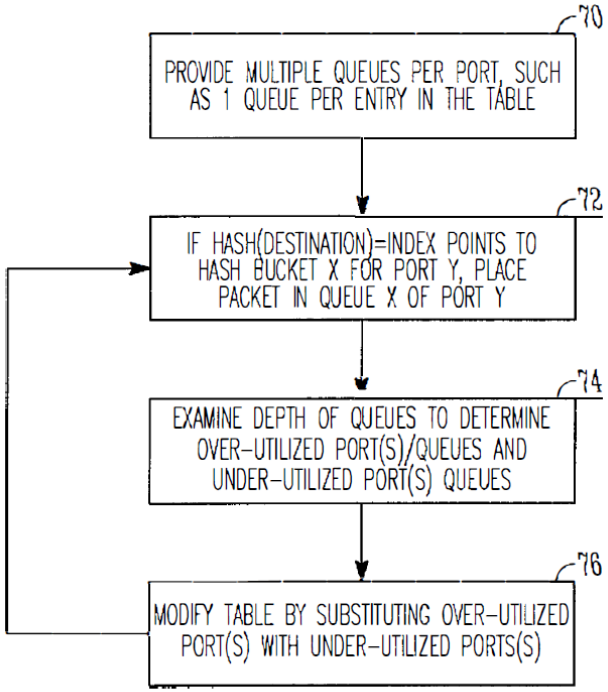
| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li ’914 at Figure 2 |

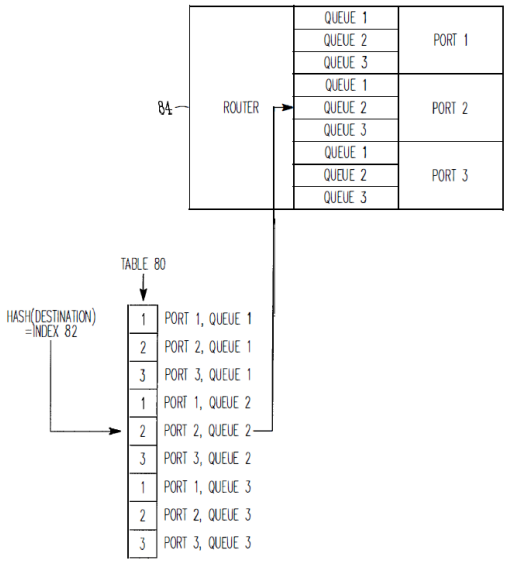
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of |

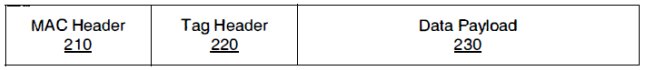
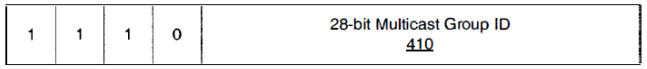
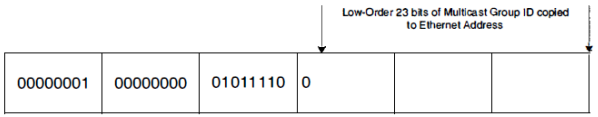
| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.”)<br><br>Li ’914 at 5:13-28 (“In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.”)<br><br>Li ’914 at 5:29-41 (“Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.”) |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).") |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Borgione ’125 at 3:30-49 (“As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.”)<br><br>Borgione ’125 at 4:3-30 (“Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.”)<br><br>Borgione ’125 at 5:28-54 (“The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.”)<br><br>Borgione ’125 at 7:1-5 (“A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).”)<br><br>Borgione ’125 at Figure 2-5 |

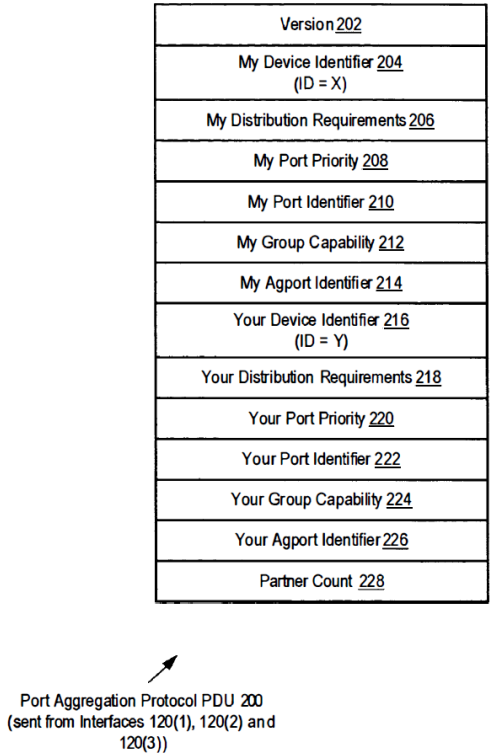
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |
| 4[f] | at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. | The Reference discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125. |

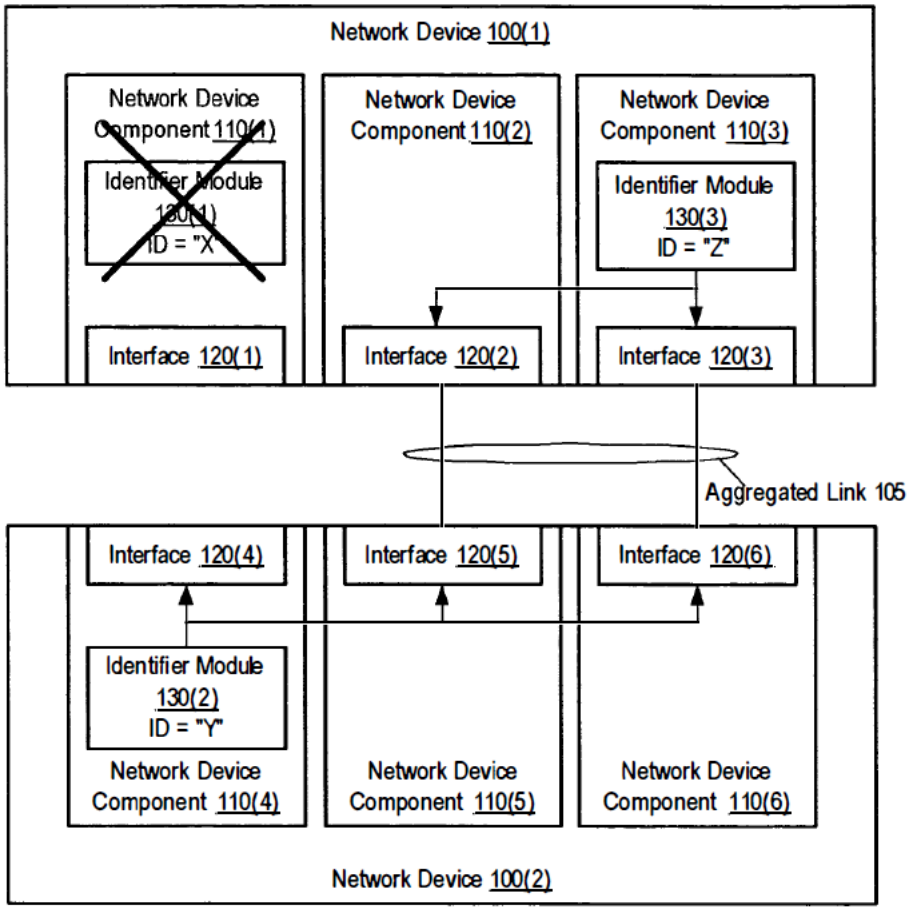
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Below is an example.<br><br>Smith '430 at 5:51-64 ("The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be oper-ated as a single logical link, referred to herein as a virtual link bundle. Network device 120(1) operates the two links cou-pling network device 120(1) to virtual network device 202 as 55 a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corre-sponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 60 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, virtual link bundles 250(1) and 250(2) are each operated as an EtherChannel™ or as an aggregated link (as described in IEEE 802.3).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the link aggregation technique. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Abstract ("Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to |

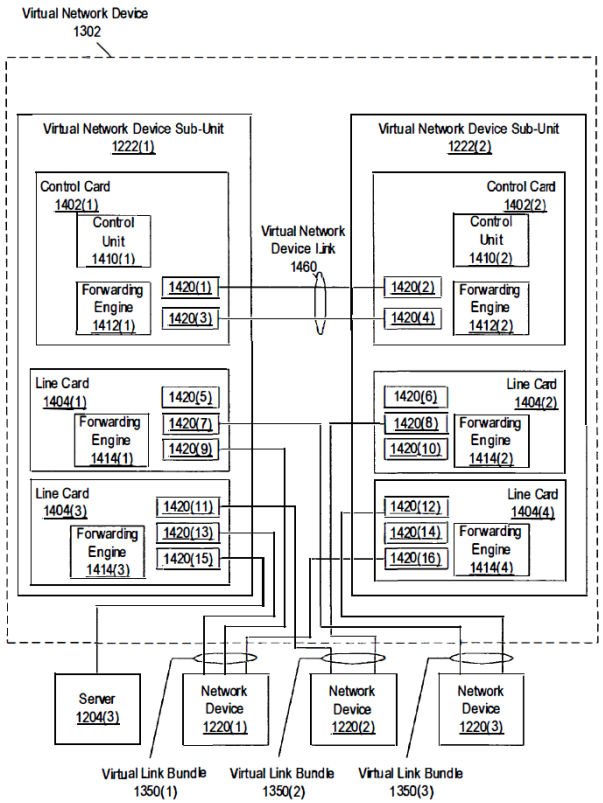
| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.”)<br><br>DeJager ’424 at 1:38-50 (“One way to relieve this bottle-neck is to provide a logical grouping of multiple ports into a single port. The bandwidth of the new port is increased since it has multiple lines (cables) connecting a switch and another network device, each line capable of carrying data at the same rate as the line connecting data sources to the switch. This grouping of ports is sometimes referred to as a port aggregation or port group. One example of such a port aggregation implementation is Cisco Technology, Inc.'s Fast EtherChannel™ port group in a Fast Ethernet network. Further information regarding Fast EtherChannel™ may be found on Cisco Technology, Inc.'s World Wide Web site www.cisco.com. This information is incorporated by reference herein for all purposes.”)<br><br>DeJager ’424 at 2:47-65 (“The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.”)<br><br>DeJager ’424 at 2:67-3:15 (“In one aspect, the present invention provides a method of distributing traffic over a network port group. The method involves receiving a packet of data to be forwarded, deter-mining a stream ID for the packet, and determining whether a prior packet having that stream ID has been distributed to a queue on a port in the group |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | during a predetermined time interval. Where a prior packet having that stream ID has not been distributed to a queue on a port of the group during the predetermined time interval, the method involves allocating the packet to a queue of a port having a lesser load in its queue than a queue of any other port of the group. The method may also involve, where a prior packet having that stream ID has been distributed to a queue on a port of the group during the predetermined time interval, allocating the packet to that queue. In addition, the method may involve monitoring the port group queues to maintain proper iden-tification of the least utilized queue.”)<br><br>DeJager ’424 at 3:16-38 (“In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.”)<br><br>DeJager ’424 at 4:29-46 (“The present invention provides methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present inven-tion is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other network environments.")<br><br>DeJager '424 at 4:47-58 ("FIG. 1 illustrates a block diagram of a simple network. The network 100 includes two servers S1. and S2, respectively, and two switches, X1 and X2, respectively, as well as four clients C1, C2, C3 and C4, respectively. Clients C , , and C4 are connected to switch X1 by, for example, Fast Ethernet links 102 via ports 1, 2, 3 and 4, respectively. Server S1 is connected to switch X1 via a port aggregation 104, which is a port group composed of ports 5 and 6 of switch X1 . Switch X1 is connected to switch X2 via a second port aggregation 106 which includes ports 7, 8 and 9. Switch X2 is connected to server S2 via port O and Fast Ethernet link 108.")<br><br>Dontu at Abstract ("Various methods and systems for preventing erroneous link aggregation due to component relocation are disclosed. Such methods include a method for changing the identifier used by a network device and communicating the identifier change to a peer network device without disrupting an aggregated link. In one embodiment, a method involves detecting an identifier change and sending a Port Aggrega-tion Protocol (PAgP) protocol data unit (PDU) that includes a new identifier and information. The information indicates the identifier change. The new identifier identifies a network device subsequent to the identifier change. Another embodi-ment of a method involves detecting an identifier change and, subsequent to the identifier change, sending a link aggregation protocol PDU that includes an "old device identifier" field dedicated to conveying an old identifier. The old identifier identifies a network device prior to the iden-tifier change.")<br><br>Dontu at Figure 2 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Dontu at Figure 3 |

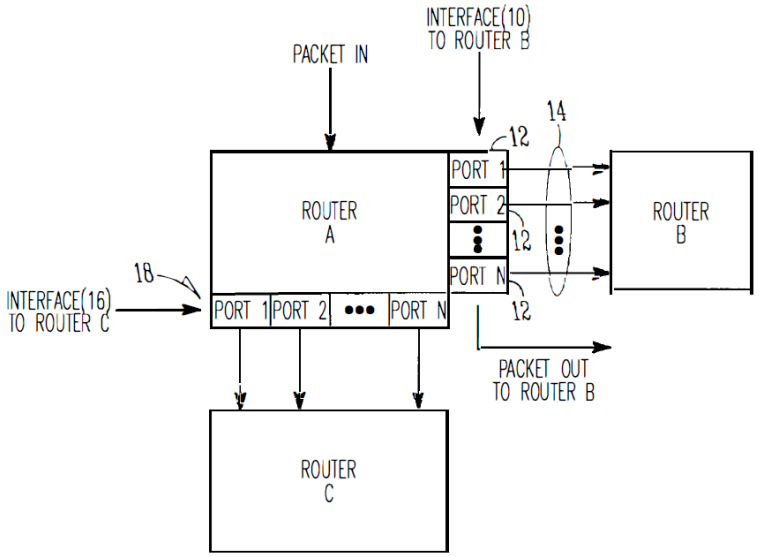
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Dontu at Figure 14 |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | |  <br>FIG. 14<br><br>Dontu at [0004] ("Link aggregation is used to logically combine two or more individual links into a single aggregated link. Link aggregation can provide improved performance and increased fault tolerance. Improved performance arises because the aggregated link appears to have a bandwidth equal to the combined bandwidth of the individual links. Traffic can be load-balanced among the individual links. Increased fault tolerance is provided since one or more individual links within an aggregated link can fail without disrupting communication between the devices coupled by the aggregated link. Link aggregation techniques include Link Aggregation Control Protocol (LACP), which is defined in IEEE 803.2ad, and Port |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | Aggregation Protocol (PAgP), which is a standard promulgated by CISCO SYS-TEMS, INC.")<br><br>Dontu at [0012] ("The method can also involve detecting whether a partner interface is executing a compatible version of PAgP. If the partner interface is not executing the compatible version of PAgP, the compatible version of PAgP can be provided to the partner interface. Alternatively, if the partner interface is not executing the compatible version of PAgP, the partner interface can be inhibited from including a link in an aggregated link.")<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).")<br><br>Dontu at [0037] ("In this example, the network devices 100(1) and 100(2) use Port Aggregation Protocol (PAgP) to form aggre-gated links. Network devices 100(1) each send PAgP pro-tocol data units (PDUs) to each other in order to determine whether any of the |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | links between the two network devices can be combined into an aggregated link. Each PAgP PDU includes an identifier that uniquely identifies the network device that sent that PAgP PDU. Within network device 100(1), identifier module 130(1) of network device compo-nent 110(1) supplies an identifier "X" to each of the inter-faces 120(1)-120(3) within network device 100(1). Inter-faces 120(1)-120(3) include identifier X in each PAgP PDU sent by those interfaces. Similarly, identifier module 130(2) of network device component 110( 4) supplies an identifier "Y" to each interface 120( 4)-120( 6) of network device 100(2). Interfaces 120( 4)-120( 6) include identifier Yin each PAgP PDU sent by those interfaces.")<br><br>Dontu at [0040] ("FIG. 2 illustrates some of the fields that can be included in a PAgP PDU. As shown, PDU 200 includes Version field 202, My Device Identifier field 204 ("My" refers to the device sending the PAgP PDU), My Distribu-tion Requirements field 206, My Port Priority field 208, My Port Identifier field 212, My Group Capability field 212, My Agport (Aggregated Port) Identifier field 214, Your Device Identifier field 216 ("Your" refers to the device to which the PAgP PDU is being sent), Your Distribution Requirements field 218, Your Port Priority field 220, Your Port Identifier field 222, Your Group Capability field 224, Your Agport Identifier field 226, and Partner Count field 228.")<br><br>Dontu at [0110] ("Interfaces 1420(13), 1420(9), and 1420(16), which are each coupled to network device 1220(1) by virtual link bundle 1350(1), form an interface bundle (e.g., an Ether-Channel (TM) port bundle). Similarly, interfaces 1420(11) and 1420(8) form another interface bundle that is coupled to network device 1220(2) by virtual link bundle 1350(2). Interfaces 1420(7) and 1420(12) form a third interface bundle that is coupled to network device 1220(3) by virtual link bundle 1350(3). Within virtual network device 1302, each interface in the same interface bundle is assigned the same logical identifier. For example, interfaces 1420(13), 1420(9), and 1420(16) are each assigned the same logical identifier. In some embodiments, packets received via one of these interfaces are tagged or otherwise associated with the logical identifier to indicate that those packets were received via the virtual link bundle coupling virtual network device 1302 to network device 1220(1). It is noted that similar interface bundles are implemented within each network device 1220(1)-1220(3), and that interfaces included in such bundles are also assigned the same logical identifier by each network device ( or by virtual network device 1302, in embodiments in |

| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | which virtual network device 1302 controls the configuration of the network devices 1220(1)-1220(3)). For example, network device 1220(1) can assign the same logical identifier to each of the interfaces coupled to virtual link bundle 1350(1).")<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 2:6-22 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.")<br><br>Li '914 at 4:9-25 ("Referring to FIG. 1, a Router A is shown having an inter-face 10 with a plurality of ports which connect Router A with Router B over a plurality of connections, lines, wires, links or bundled links 14. The ports 12 of Router A are configured to permit transmission of packets from Router A to Router B, and these ports 12 may be referred to as output ports, egress ports, links, or the like. As shown in FIG. 1, port 1 to port N may be |

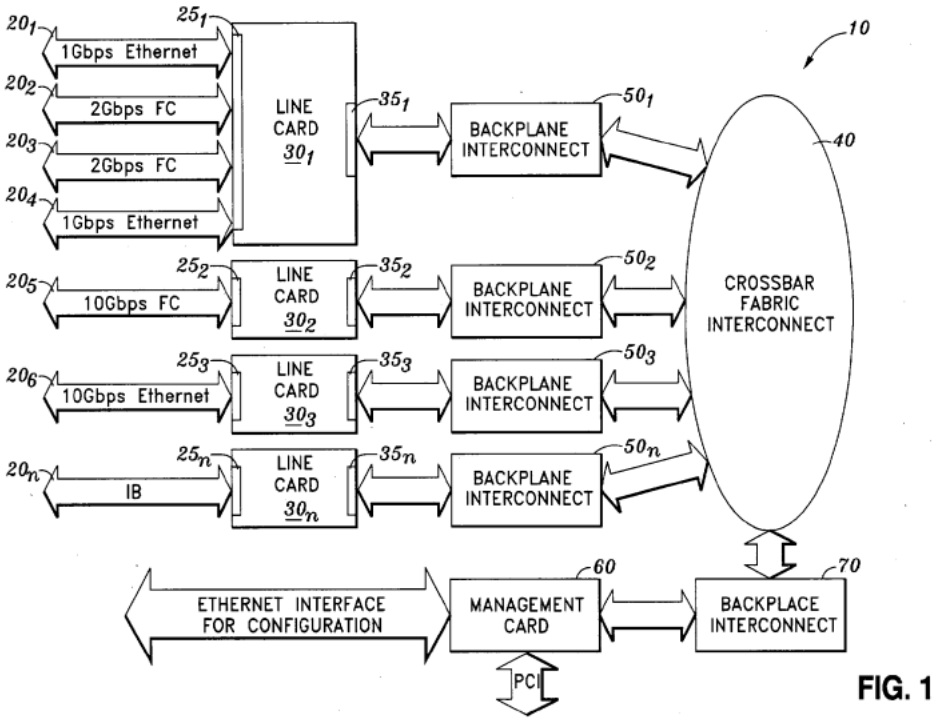
| No. | '740 Patent Claim 4 | The Reference |
|---|---|---|
| | | connected with Router B, and there may be additional interfaces 16 having ports 18 connected with other routers in the network. When a packet is received by Router A, Router A determines whether the received packet should be transmitted to Router B or to other routers connected to Router A, based in part upon the destination address of the packet. If a packet is to be transmitted from Router A to Router B, then Router A may transmit this packet over the one of the ports 12 shown in FIG. 1.")<br><br>Li '914 at Figure 1<br>  <br><br>*FIG. 1*<br><br>Borgione '125 at 1:55-65 ("Link nodes 110 and 120 can be in physically remote locations, thereby connecting their associated local area networks (LANs). The plurality of network links 150 between link nodes 110 and 120 can be aggregated as a single logical link over which all traffic between link nodes 110 and 120 is distributed. Such aggregation multiplies the available bandwidth for communications between link nodes 110 and 120, and therefore |

| No. | ’740 Patent Claim 4 | The Reference |
|---|---|---|
| | | between the two local area networks. When appropriately configured, such a connection can permit the two local area networks to interact as if they were one large local area network.”)<br><br>Borgione ’125 at 1:66-2:7 (“As stated above, the plurality of network links between 110 and 120 can be aggregated as a single logical link. In this manner, each link node 110 and 120 sees the plurality of network links between them as one logical interface. One type of such an aggregate of links is an EtherChannel, a protocol that allows up to eight Fast Ethernet or Gigabit Ethernet links to be aggregated. Routing protocols treat the aggregated links as a single, routed interface with a common IP address.”)<br><br>Borgione ’125 at 5:28-50 (“The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.”) |

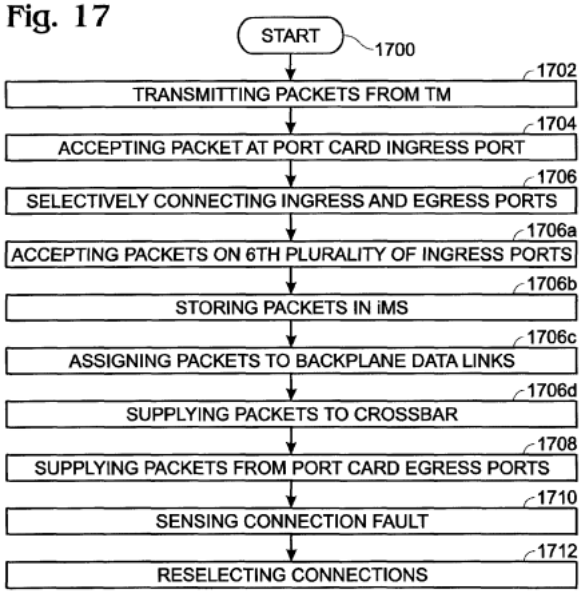
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| 5[preamble] | A method for communication, comprising: | The Reference discloses a method for communication.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 5[a] | coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel; | The Reference discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 5[b] | coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel; | The Reference discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 5[c] | receiving a data frame having frame attributes sent between the communication network and the network node: | The Reference discloses receiving a data frame having frame attributes sent between the communication network and the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 5[d] | selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and | The Reference discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified |

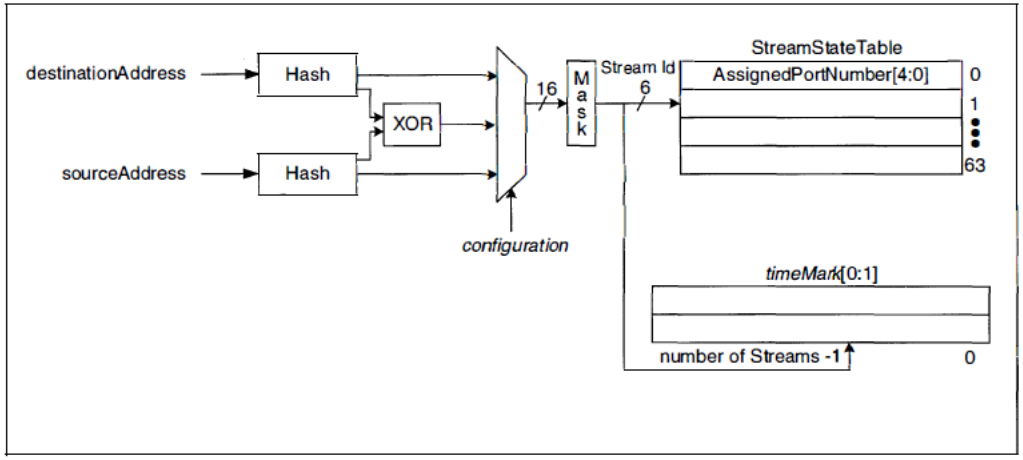
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

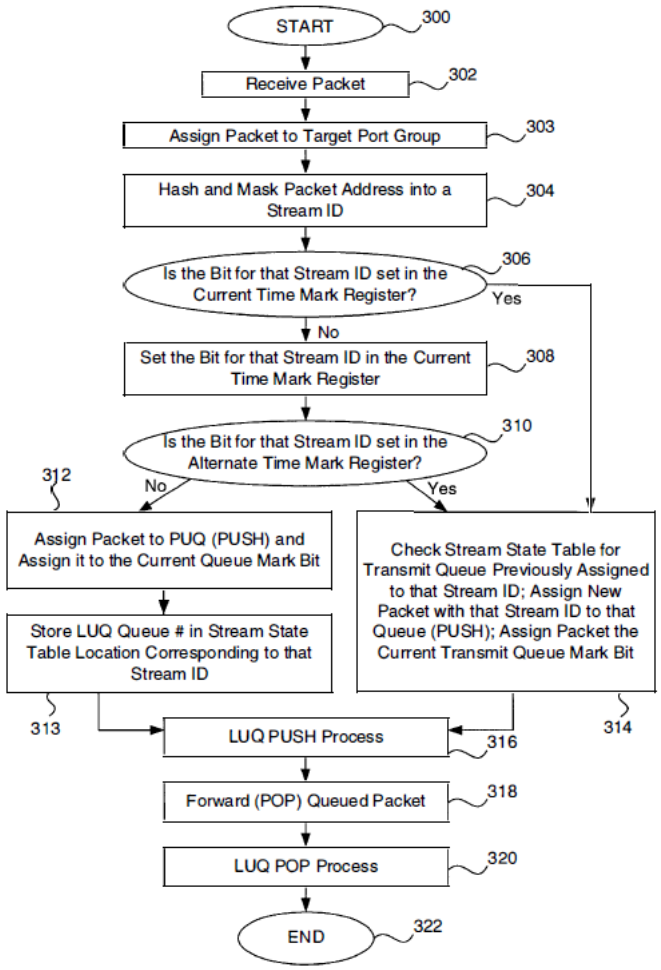
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including selecting physical links over which to send a packet. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and |

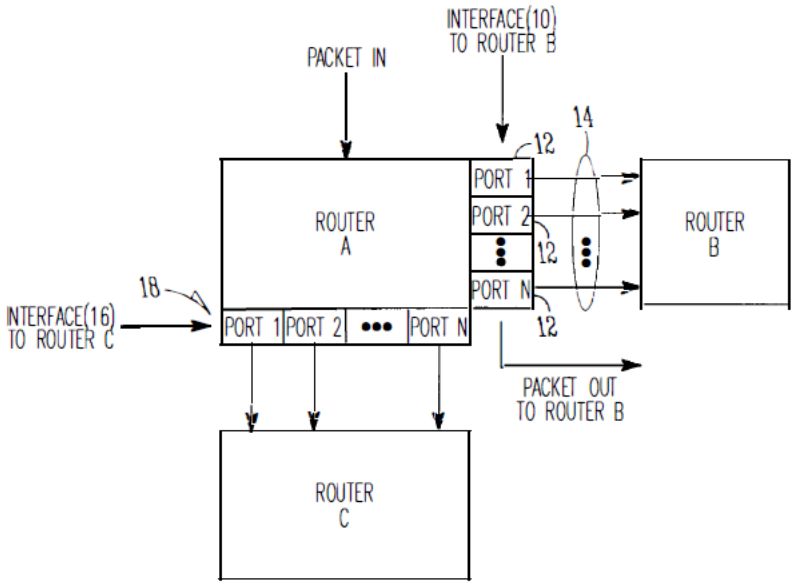
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

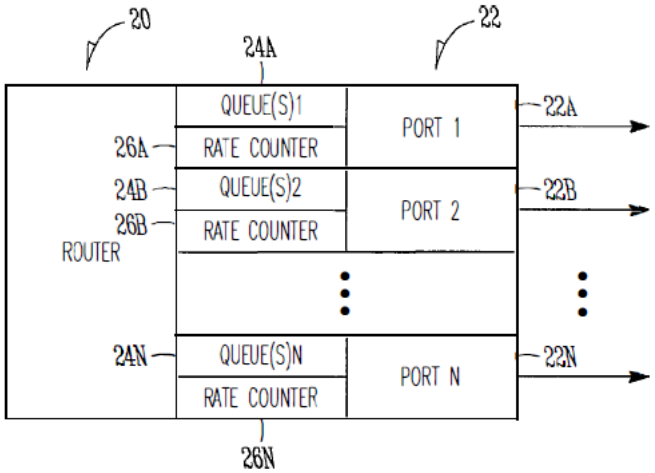
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

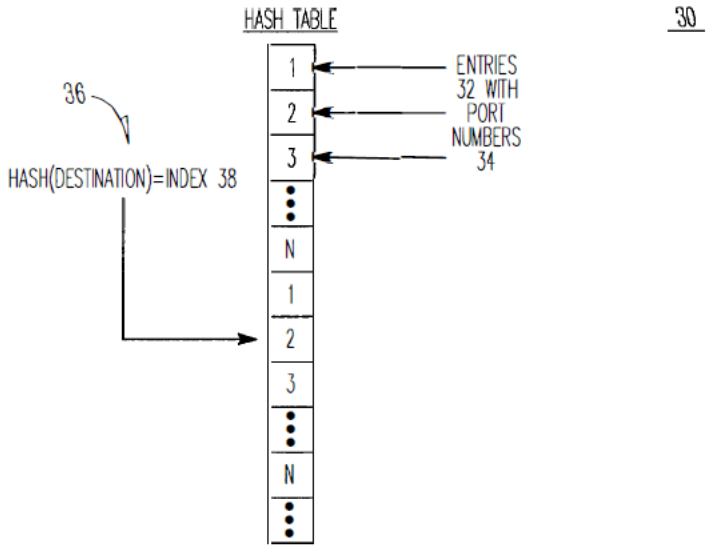
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub- |

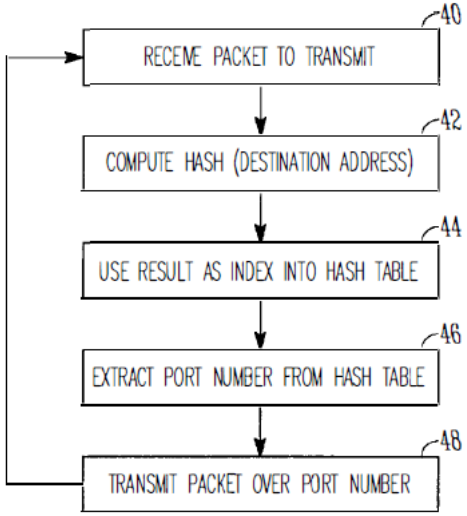
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.") |

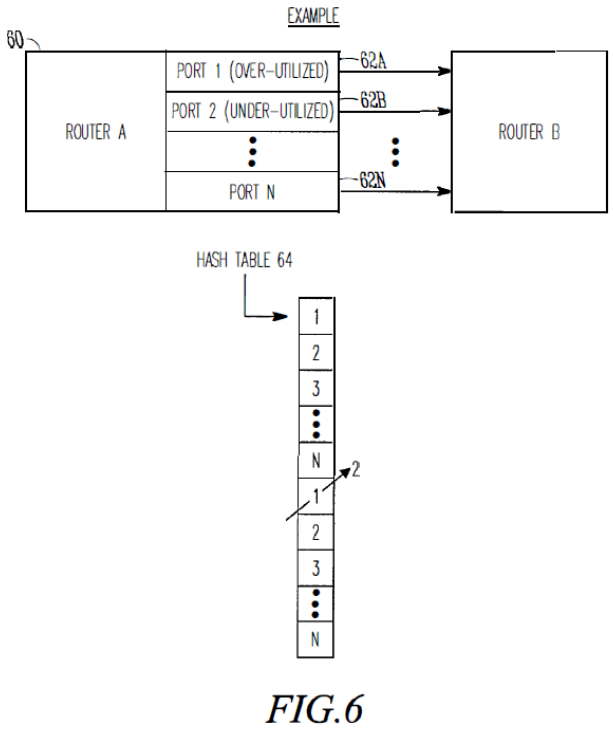
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

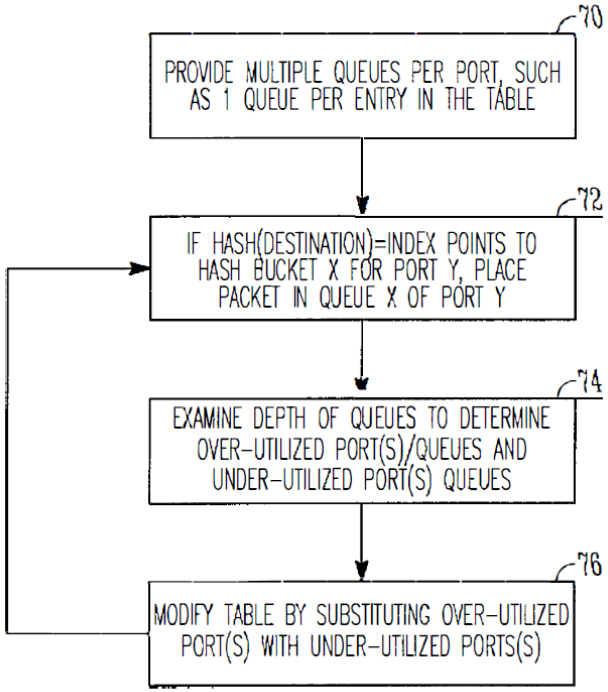
| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li ’914 at Figure 2 |

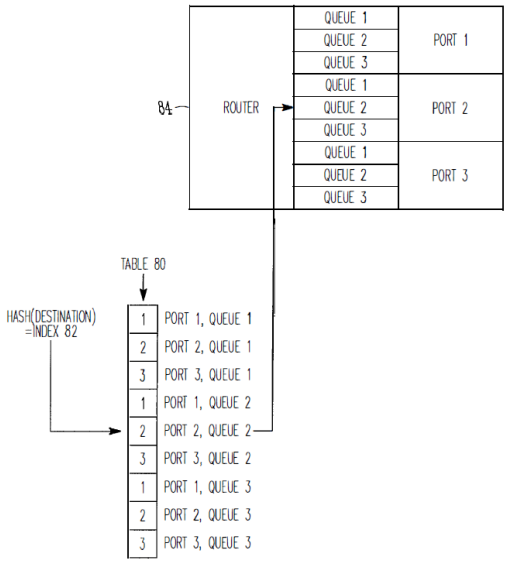
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router |

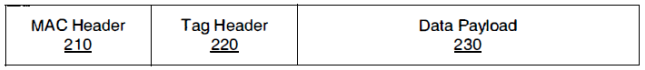
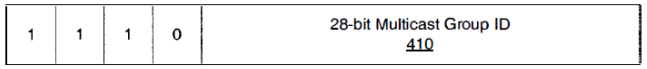
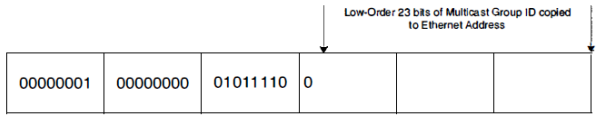
| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

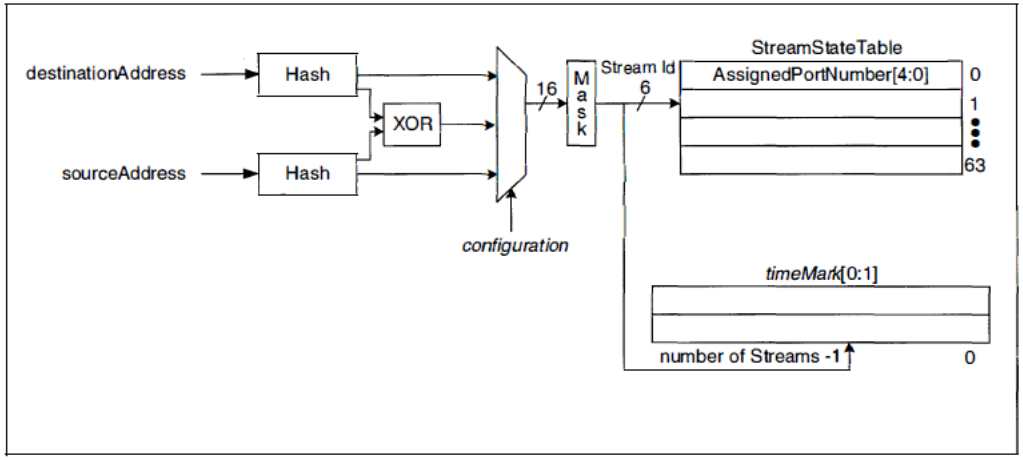
| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.”)<br><br>Li ’914 at 5:13-28 (“In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.”)<br><br>Li ’914 at 5:29-41 (“Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.”) |

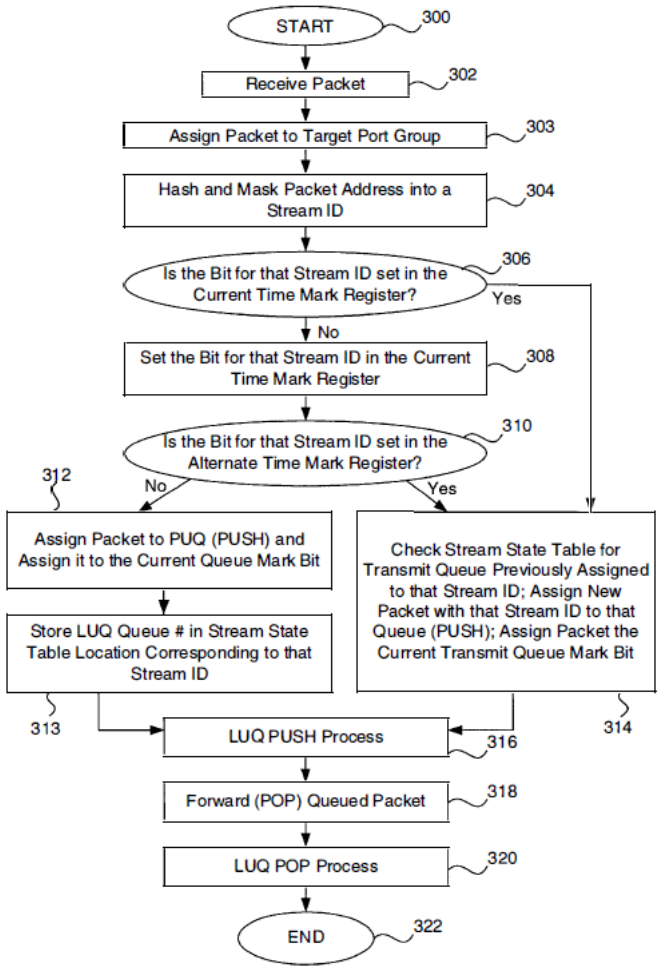
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).") |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.")<br><br>Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |
| 5[e] | sending the data frame over the selected first and second physical links, | The Reference discloses sending the data frame over the selected first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and |

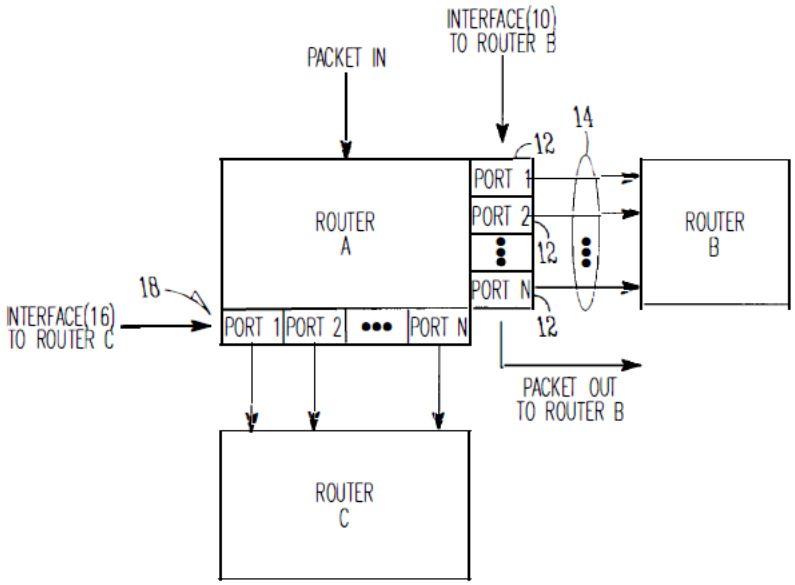
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

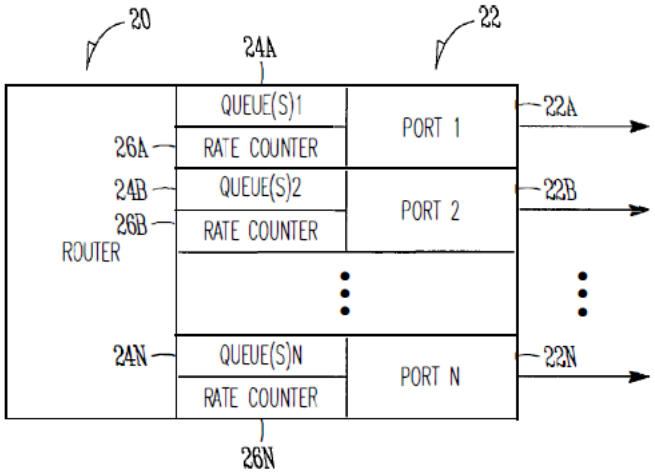
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

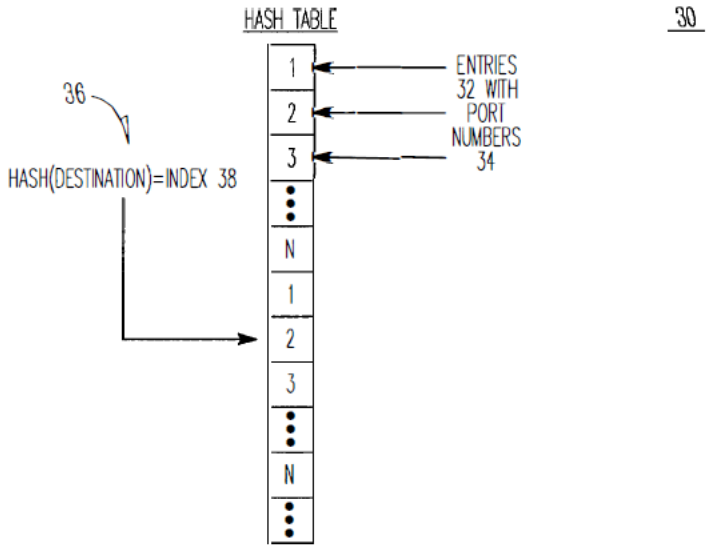
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub- |

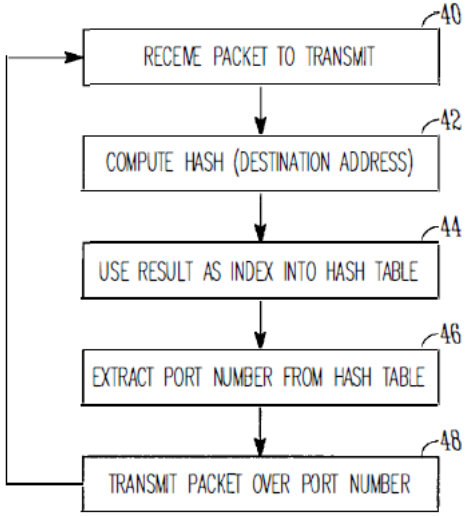
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.") |

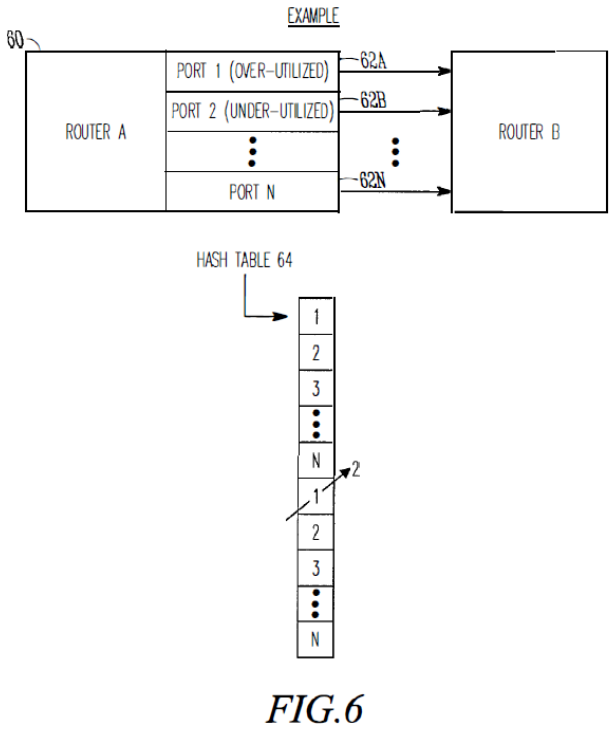
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

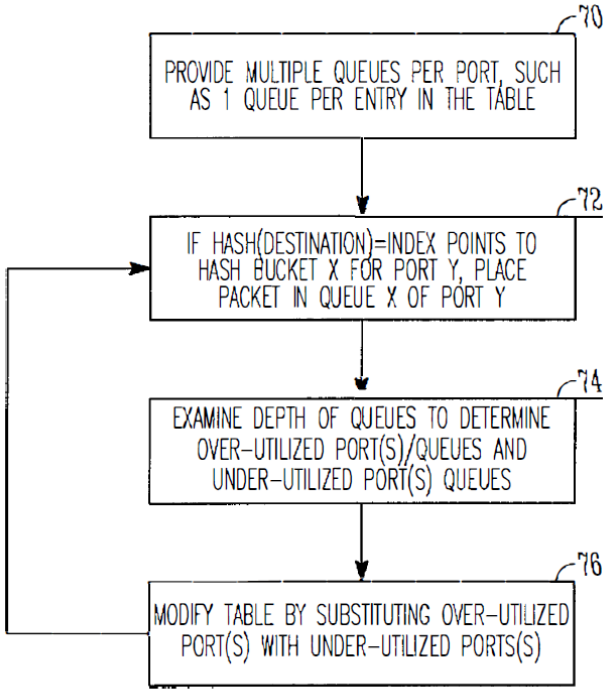
| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li ’914 at Figure 2 |

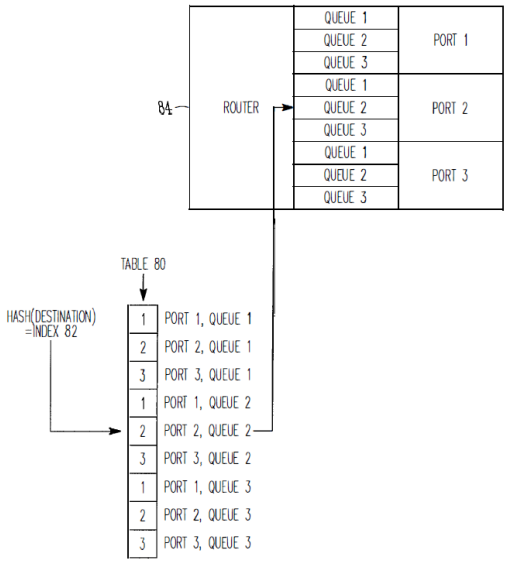
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router |

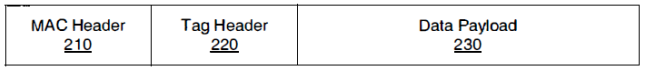
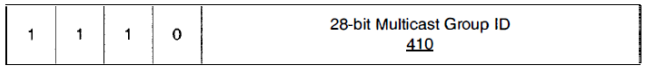
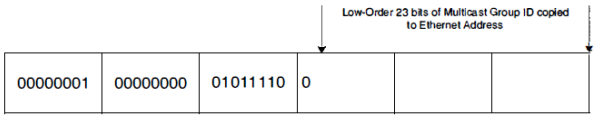
| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.”)<br><br>Li ’914 at 5:13-28 (“In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.”)<br><br>Li ’914 at 5:29-41 (“Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.”) |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).") |

| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Borgione ’125 at 3:30-49 (“As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.”)<br><br>Borgione ’125 at 4:3-30 (“Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.")<br><br>Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5 |

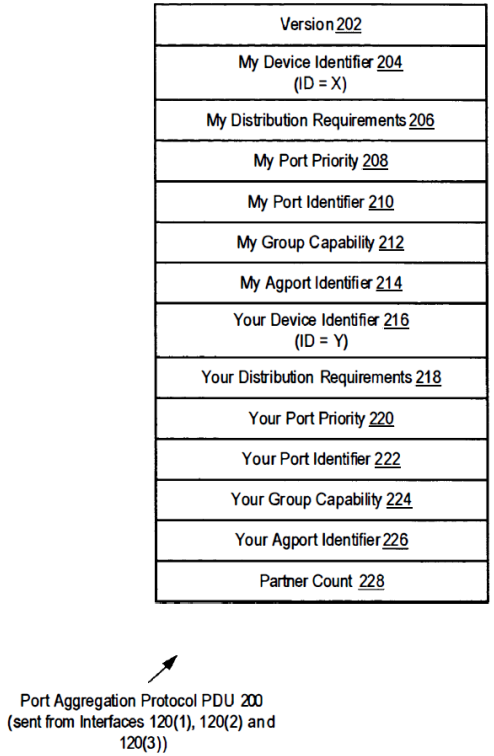
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |
| 5[f] | coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase | The Reference discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

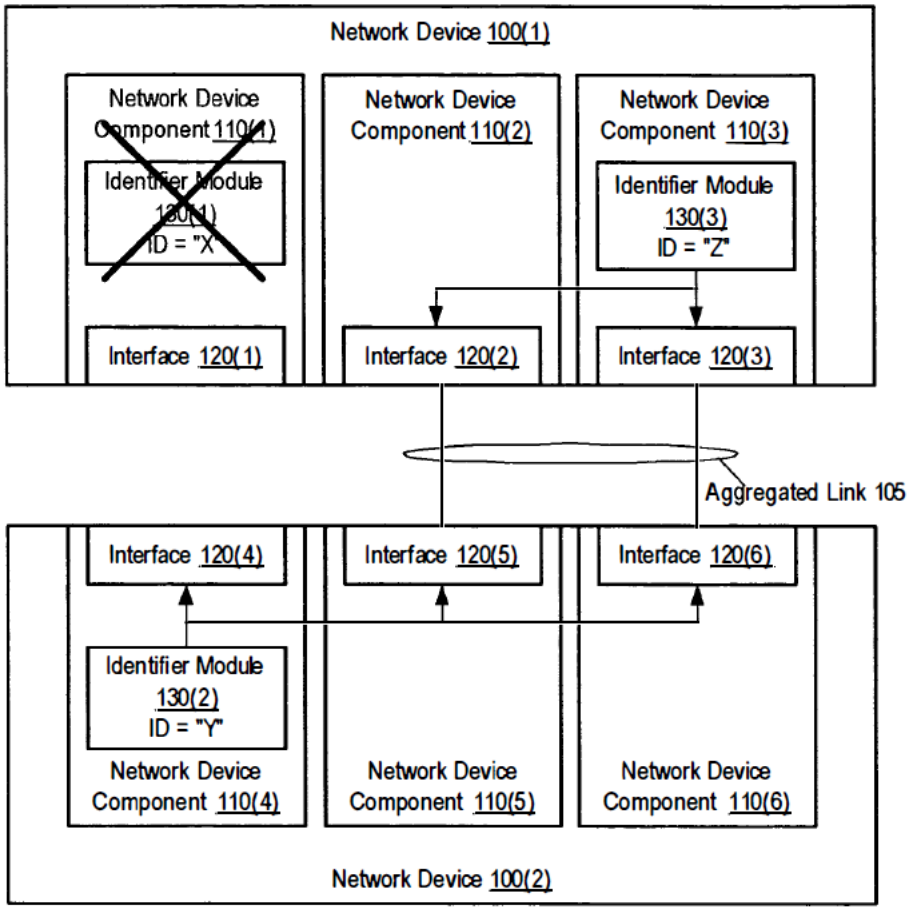
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | a data bandwidth provided to the network node. | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below is an example.<br><br>Smith '430 at 5:51-64 ("The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be oper-ated as a single logical link, referred to herein as a virtual link bundle. Network device 120(1) operates the two links cou-pling network device 120(1) to virtual network device 202 as 55 a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corre-sponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 60 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, virtual link bundles 250(1) and 250(2) are each operated as an EtherChannel™ or as an aggregated link (as described in IEEE 802.3).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the link aggregation technique. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Abstract ("Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This |

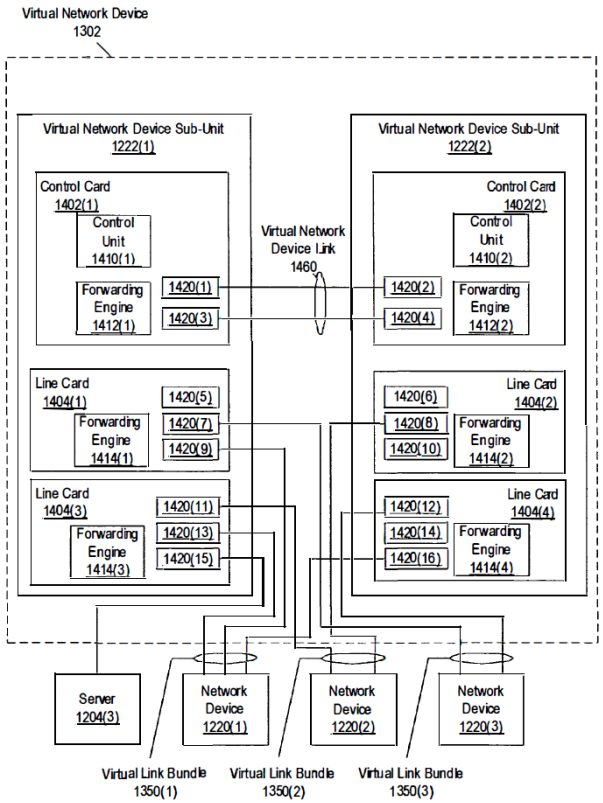
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.")<br><br>DeJager '424 at 1:38-50 ("One way to relieve this bottle-neck is to provide a logical grouping of multiple ports into a single port. The bandwidth of the new port is increased since it has multiple lines (cables) connecting a switch and another network device, each line capable of carrying data at the same rate as the line connecting data sources to the switch. This grouping of ports is sometimes referred to as a port aggregation or port group. One example of such a port aggregation implementation is Cisco Technology, Inc.'s Fast EtherChannel™ port group in a Fast Ethernet network. Further information regarding Fast EtherChannel™ may be found on Cisco Technology, Inc.'s World Wide Web site www.cisco.com. This information is incorporated by reference herein for all purposes.")<br><br>DeJager '424 at 2:47-65 ("The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.")<br><br>DeJager '424 at 2:67-3:15 ("In one aspect, the present invention provides a method of distributing traffic over a network port group. The method involves receiving a packet of |

| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | data to be forwarded, deter-mining a stream ID for the packet, and determining whether a prior packet having that stream ID has been distributed to a queue on a port in the group during a predetermined time interval. Where a prior packet having that stream ID has not been distributed to a queue on a port of the group during the predetermined time interval, the method involves allocating the packet to a queue of a port having a lesser load in its queue than a queue of any other port of the group. The method may also involve, where a prior packet having that stream ID has been distributed to a queue on a port of the group during the predetermined time interval, allocating the packet to that queue. In addition, the method may involve monitoring the port group queues to maintain proper iden-tification of the least utilized queue.”)<br><br>DeJager ’424 at 3:16-38 (“In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.”)<br><br>DeJager ’424 at 4:29-46 (“The present invention provides methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present inven-tion is preferably dynamic, that is, packets from a given stream may be |

| No. | ’740 Patent Claim 5 | The Reference |
|---|---|---|
| | | forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other network environments.”)<br><br>DeJager ’424 at 4:47-58 (“FIG. 1 illustrates a block diagram of a simple network. The network 100 includes two servers S1. and S2, respectively, and two switches, X1 and X2, respectively, as well as four clients C1, C2, C3 and C4, respectively. Clients C , , and C4 are connected to switch X1 by, for example, Fast Ethernet links 102 via ports 1, 2, 3 and 4, respectively. Server S1 is connected to switch X1 via a port aggregation 104, which is a port group composed of ports 5 and 6 of switch X1 . Switch X1 is connected to switch X2 via a second port aggregation 106 which includes ports 7, 8 and 9. Switch X2 is connected to server S2 via port O and Fast Ethernet link 108.”)<br><br>Dontu at Abstract (“Various methods and systems for preventing erroneous link aggregation due to component relocation are disclosed. Such methods include a method for changing the identifier used by a network device and communicating the identifier change to a peer network device without disrupting an aggregated link. In one embodiment, a method involves detecting an identifier change and sending a Port Aggrega-tion Protocol (PAgP) protocol data unit (PDU) that includes a new identifier and information. The information indicates the identifier change. The new identifier identifies a network device subsequent to the identifier change. Another embodi-ment of a method involves detecting an identifier change and, subsequent to the identifier change, sending a link aggregation protocol PDU that includes an "old device identifier" field dedicated to conveying an old identifier. The old identifier identifies a network device prior to the iden-tifier change.”)<br><br>Dontu at Figure 2 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Dontu at Figure 3 |

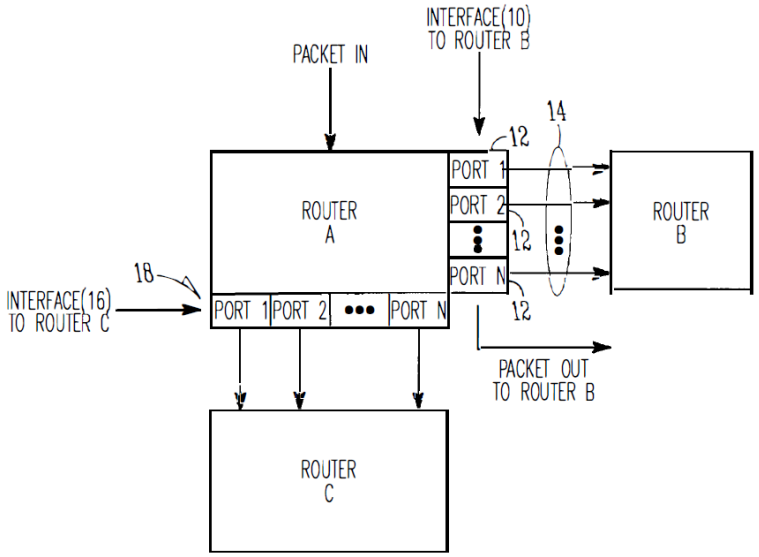
| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Dontu at Figure 14 |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | |  <br>FIG. 14<br><br>Dontu at [0004] ("Link aggregation is used to logically combine two or more individual links into a single aggregated link. Link aggregation can provide improved performance and increased fault tolerance. Improved performance arises because the aggregated link appears to have a bandwidth equal to the combined bandwidth of the individual links. Traffic can be load-balanced among the individual links. Increased fault tolerance is provided since one or more individual links within an aggregated link can fail without disrupting communication between the devices coupled by the aggregated link. Link aggregation techniques include Link Aggregation Control Protocol (LACP), which is defined in IEEE 803.2ad, and Port |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | Aggregation Protocol (PAgP), which is a standard promulgated by CISCO SYS-TEMS, INC.")<br><br>Dontu at [0012] ("The method can also involve detecting whether a partner interface is executing a compatible version of PAgP. If the partner interface is not executing the compatible version of PAgP, the compatible version of PAgP can be provided to the partner interface. Alternatively, if the partner interface is not executing the compatible version of PAgP, the partner interface can be inhibited from including a link in an aggregated link.")<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).")<br><br>Dontu at [0037] ("In this example, the network devices 100(1) and 100(2) use Port Aggregation Protocol (PAgP) to form aggre-gated links. Network devices 100(1) each send PAgP pro-tocol data units (PDUs) to each other in order to determine whether any of the |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | links between the two network devices can be combined into an aggregated link. Each PAgP PDU includes an identifier that uniquely identifies the network device that sent that PAgP PDU. Within network device 100(1), identifier module 130(1) of network device compo-nent 110(1) supplies an identifier "X" to each of the inter-faces 120(1)-120(3) within network device 100(1). Inter-faces 120(1)-120(3) include identifier X in each PAgP PDU sent by those interfaces. Similarly, identifier module 130(2) of network device component 110( 4) supplies an identifier "Y" to each interface 120( 4)-120( 6) of network device 100(2). Interfaces 120( 4)-120( 6) include identifier Yin each PAgP PDU sent by those interfaces.")<br><br>Dontu at [0040] ("FIG. 2 illustrates some of the fields that can be included in a PAgP PDU. As shown, PDU 200 includes Version field 202, My Device Identifier field 204 ("My" refers to the device sending the PAgP PDU), My Distribu-tion Requirements field 206, My Port Priority field 208, My Port Identifier field 212, My Group Capability field 212, My Agport (Aggregated Port) Identifier field 214, Your Device Identifier field 216 ("Your" refers to the device to which the PAgP PDU is being sent), Your Distribution Requirements field 218, Your Port Priority field 220, Your Port Identifier field 222, Your Group Capability field 224, Your Agport Identifier field 226, and Partner Count field 228.")<br><br>Dontu at [0110] ("Interfaces 1420(13), 1420(9), and 1420(16), which are each coupled to network device 1220(1) by virtual link bundle 1350(1), form an interface bundle (e.g., an Ether-Channel (TM) port bundle). Similarly, interfaces 1420(11) and 1420(8) form another interface bundle that is coupled to network device 1220(2) by virtual link bundle 1350(2). Interfaces 1420(7) and 1420(12) form a third interface bundle that is coupled to network device 1220(3) by virtual link bundle 1350(3). Within virtual network device 1302, each interface in the same interface bundle is assigned the same logical identifier. For example, interfaces 1420(13), 1420(9), and 1420(16) are each assigned the same logical identifier. In some embodiments, packets received via one of these interfaces are tagged or otherwise associated with the logical identifier to indicate that those packets were received via the virtual link bundle coupling virtual network device 1302 to network device 1220(1). It is noted that similar interface bundles are implemented within each network device 1220(1)-1220(3), and that interfaces included in such bundles are also assigned the same logical identifier by each network device ( or by virtual network device 1302, in embodiments in |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | which virtual network device 1302 controls the configuration of the network devices 1220(1)-1220(3)). For example, network device 1220(1) can assign the same logical identifier to each of the interfaces coupled to virtual link bundle 1350(1).")<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 2:6-22 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.")<br><br>Li '914 at 4:9-25 ("Referring to FIG. 1, a Router A is shown having an inter-face 10 with a plurality of ports which connect Router A with Router B over a plurality of connections, lines, wires, links or bundled links 14. The ports 12 of Router A are configured to permit transmission of packets from Router A to Router B, and these ports 12 may be referred to as output ports, egress ports, links, or the like. As shown in FIG. 1, port 1 to port N may be |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | connected with Router B, and there may be additional interfaces 16 having ports 18 connected with other routers in the network. When a packet is received by Router A, Router A determines whether the received packet should be transmitted to Router B or to other routers connected to Router A, based in part upon the destination address of the packet. If a packet is to be transmitted from Router A to Router B, then Router A may transmit this packet over the one of the ports 12 shown in FIG. 1.")<br><br>Li '914 at Figure 1<br>  <br><br>*FIG. 1*<br><br>Borgione '125 at 1:55-65 ("Link nodes 110 and 120 can be in physically remote locations, thereby connecting their associated local area networks (LANs). The plurality of network links 150 between link nodes 110 and 120 can be aggregated as a single logical link over which all traffic between link nodes 110 and 120 is distributed. Such aggregation multiplies the available bandwidth for communications between link nodes 110 and 120, and therefore |

| No. | '740 Patent Claim 5 | The Reference |
|---|---|---|
| | | between the two local area networks. When appropriately configured, such a connection can permit the two local area networks to interact as if they were one large local area network.")<br><br>Borgione '125 at 1:66-2:7 ("As stated above, the plurality of network links between 110 and 120 can be aggregated as a single logical link. In this manner, each link node 110 and 120 sees the plurality of network links between them as one logical interface. One type of such an aggregate of links is an EtherChannel, a protocol that allows up to eight Fast Ethernet or Gigabit Ethernet links to be aggregated. Routing protocols treat the aggregated links as a single, routed interface with a common IP address.")<br><br>Borgione '125 at 5:28-50 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.") |

| No. | '740 Patent Claim 6 | The Reference |
|---|---|---|
| 6 | The method according to claim 1, wherein | The Reference discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of |

| No. | '740 Patent Claim 6 | The Reference |
|---|---|---|
| | coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. | multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, and Wong.<br><br>Below is an example.<br><br>Wong at [0072] ("The address resolution circuit 136 comprises a port based load balancing circuit 258 including: a source port selection multiplexer 260 having a plurality of (2M=8) inputs 262 each coupled to receive a corresponding one of the port mapping values from output 256 of a corresponding one of the configuration registers 252 of unit 116, an output 264, and a control port 268 coupled to receive a three-bit source port ID value carried by a source port signal received from the input queuing control logic unit 121 (FIG. 3A) of the particular switching device; a destination trunk port register 272 having an input 274 coupled to receive selected ones of the port mapping values from output 264 of multiplexer 260, and a plurality of (2M=8) outputs 276; and a trunk port selection multiplexer 280 having a plurality of (2M=8) inputs 282 each coupled to receive a two-bit value from a corre-sponding one of the outputs 276 of register 272, an output 284, and a control port 286 coupled to receive a trunked port ID signal carrying a three bit trunked port ID value (x,x,t) from the packet routing table 134 (FIGS. 3A and 3B) as further explained below.")<br><br>Wong at [0073] ("In the depicted embodiment, the register bank 251 of unit 116 includes eight of the trunk port configuration registers 252, designated TP 0-TP 7, each being associated with a corresponding one of the eight network ports 14 of the particular switching device 12 (FIG. 3A). The source port selection multiplexer 260 selects from the outputs 256 |

| No. | '740 Patent Claim 6 | The Reference |
|---|---|---|
| | | of registers 252 in response to the source port value received at its control port 268 from the input queuing control logic 121 (FIG. 3A). The source port value indicates the source port at which a particular packet has been received. Therefore, the multiplexer 260 selects one of the registers 252 which corresponds with the source port associated with the received packet. As further described below, the 16-bit port mapping value stored in the selected one of the registers 252 includes eight separate 2-bit port values select each indicat-ing a destination port associated with the particular packet received at the corresponding source port.") |

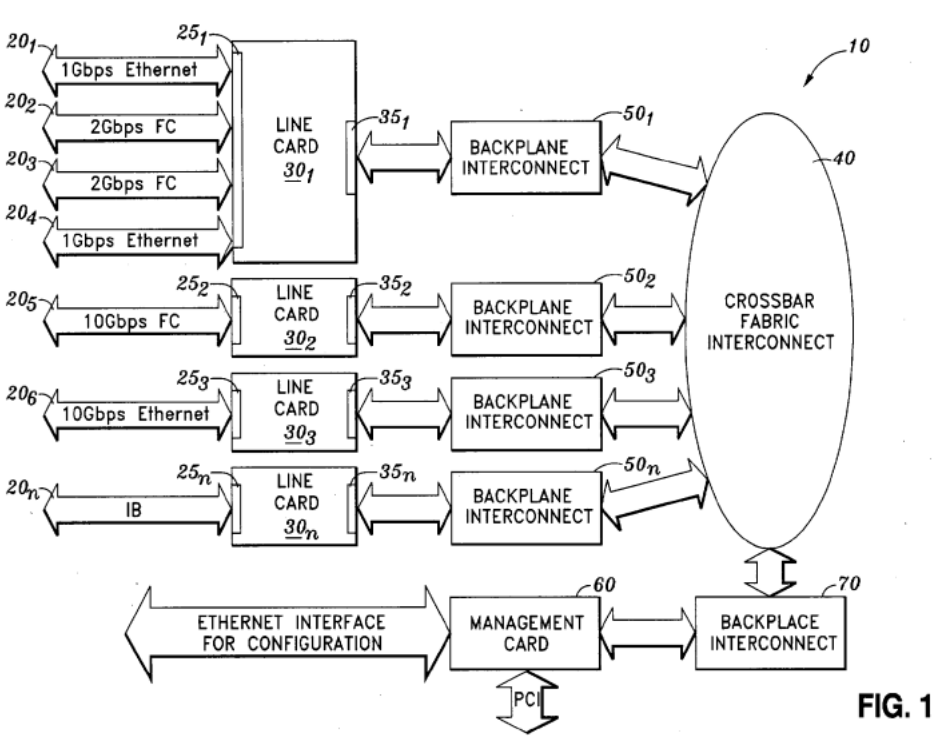
| No. | '740 Patent Claim 7 | The Reference |
|---|---|---|
| 7 | The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. | The Reference discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| 8 | The method according to claim 1, wherein selecting the first and second physical links comprises applying a | The Reference discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. |

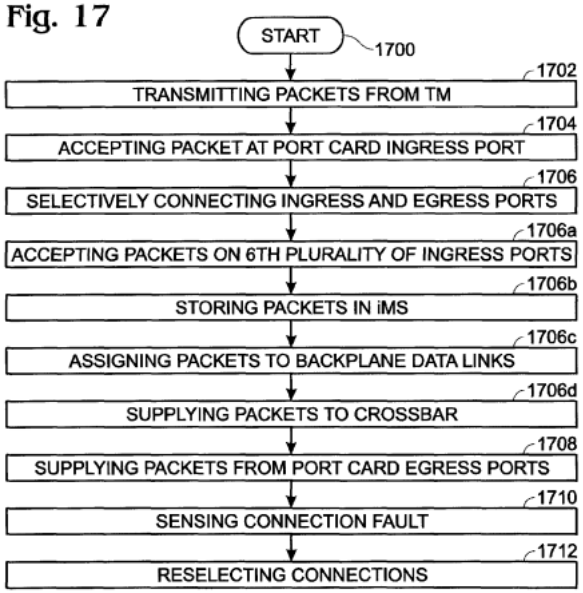
| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | mapping function to the at least one of the frame attributes. | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Solomon, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Solomon at [0024] ("In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.")<br><br>Solomon at [0048] ("The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.")<br><br>Solomon at [0059] ("In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | | function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.")<br><br>Solomon at [0060] ("As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.")<br><br>Solomon at [0065] ("Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.")<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | | of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | | ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | | ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the |

| No. | '740 Patent Claim 8 | The Reference |
|---|---|---|
| | | uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).") |

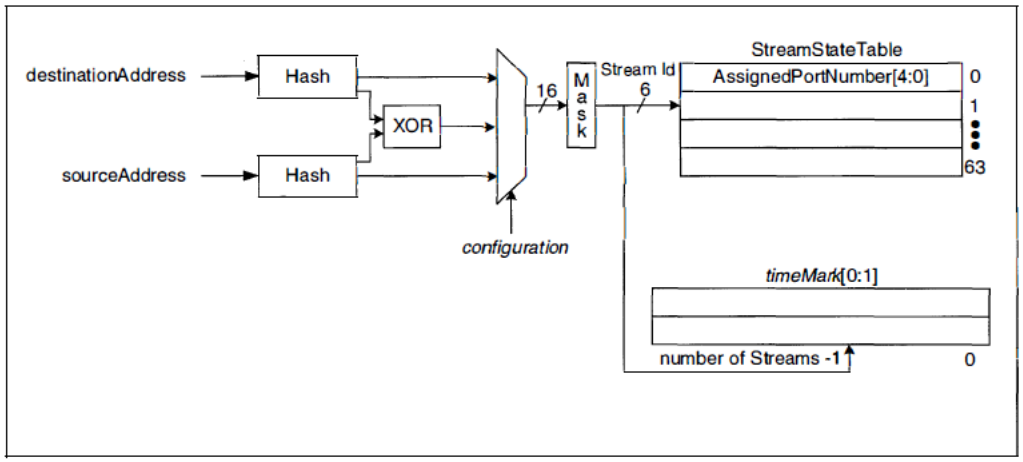
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| 9 | The method according to claim 8, wherein applying the mapping function comprises applying a hashing function. | The Reference discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

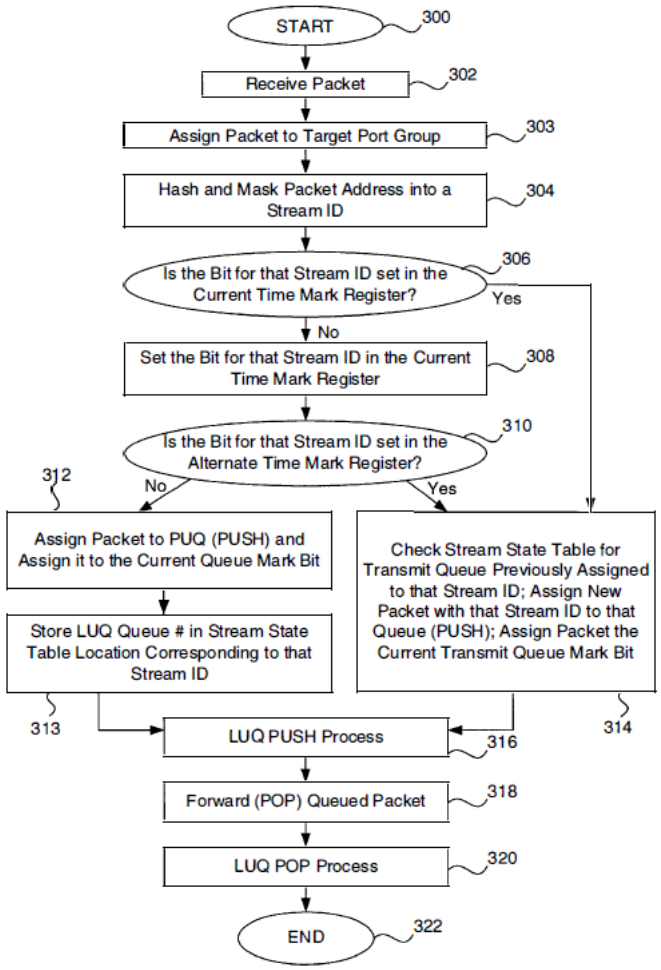
| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher ’530, Solomon, Smith ’430, Alexander, DeJager ’424, Dontu, Li ’914, and Borgione ’125.<br><br>Below are examples of such references.<br><br>Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)<br><br>Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”)<br><br>Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) |

| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | | Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”)<br><br>Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)<br><br>Smith ’430 at 10:21-39 (“The same logical identifiers can be used to identify uplink interface bundles by each of virtual network device sub-units 122(1) and 122(2), and the virtual network device sub-units coordinate to assign the same logical identifier to each uplink interface within the same uplink interface bundle. When for-warding packets via an uplink interface bundle identified by a particular logical identifier, each virtual network device sub-unit 122(1) and 122(2) generates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink inter-faces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 122(1) is forwarding a packet via the uplink interface bundle that includes interfaces 320 (9), 320(13), and 320(16), the hash value generated by virtual network device sub-unit will identify one of its interfaces 320(9) or 320(13).”) |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | Alexander at 1:36-55 ("A typical prior art Ethernet link aggregation implemen-tation utilizes a hardware means for distributing packets across multiple physical links, and re-aggregating them at the receiving end. This is typically due to the high speeds involved (100 Mb/s or even 1000 Mb/s per link) in the packet transfer. The use of such hardware is expensive in terms of the silicon resources required to perform the distribution and collection functions, and is also inflexible in terms of the algorithms used to determine how packets may be distributed across links. Additionally, the complexity of the distribution function when accounting for the various packet ordering and sequencing requirements of the Ethernet protocol renders a hardware-only approach difficult to design and debug. A well-partitioned, mixed hardware/ firmware approach is preferable when implementing link aggregation at high speeds. This approach, permits high speeds to be attained while at the same time preserving flexibility in implementation, which is necessary for track-ing changing standards or implementing different distribu-tion algorithms.")<br><br>Alexander at 2:55-67 ("Advantageously, the source context corresponding to an extracted source address is derived by producing a hash key through application of a hash function to the extracted source address. The incoming port on which the packet containing the extracted source address was received is identified. If the identified incoming port is within an aggregated grouping of incoming ports, then a port identifier representative that aggregated grouping is derived. If the identified incoming port is not within an aggregated group-ing of incoming ports, then a port identifier representative of the identified incoming port is derived. The hash key and the port identifier are then combined to form the source context corresponding to the extracted source address.")<br><br>Alexander at 3:1-40 ("The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest<br>values. |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains $2^N$ entries occupying $2^N$ consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.")<br><br>Alexander at 6:49-65 ("If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and |

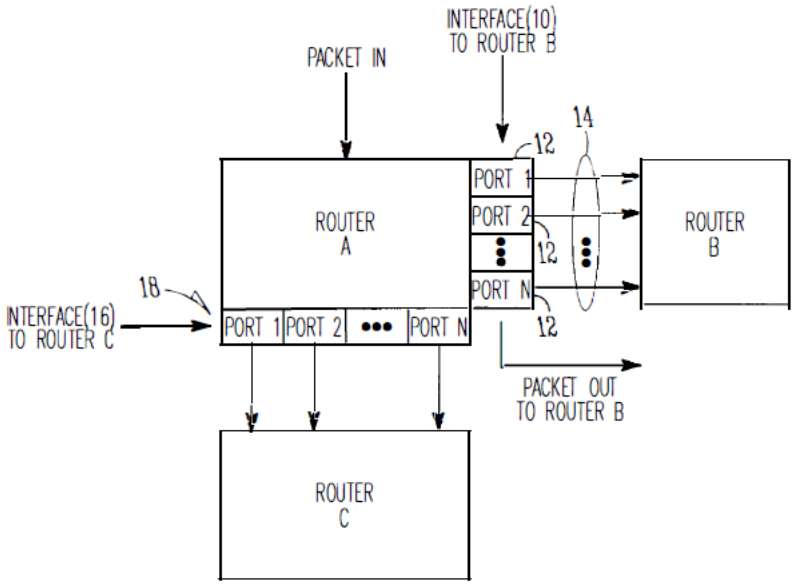
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6- |

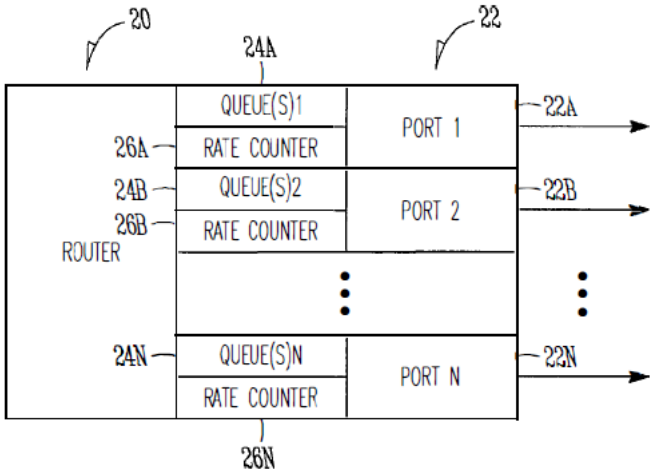
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more |

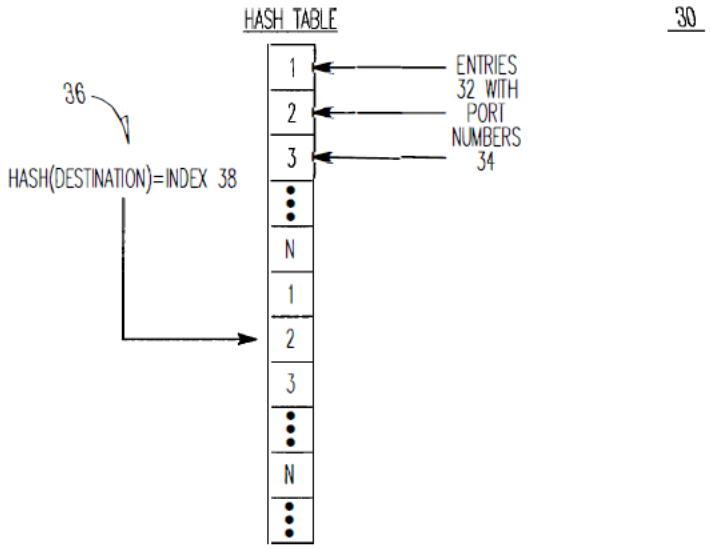
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet |

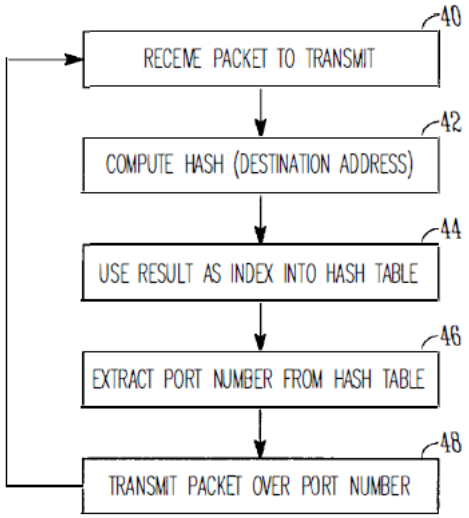
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and |

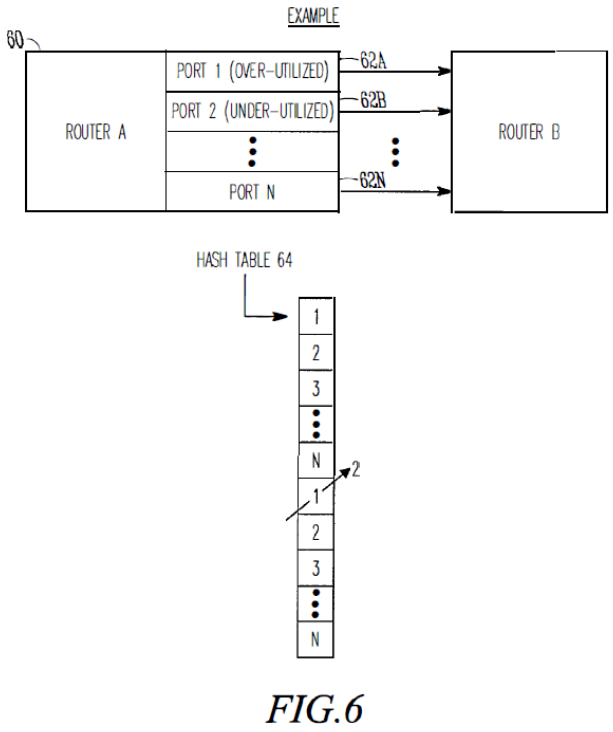
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

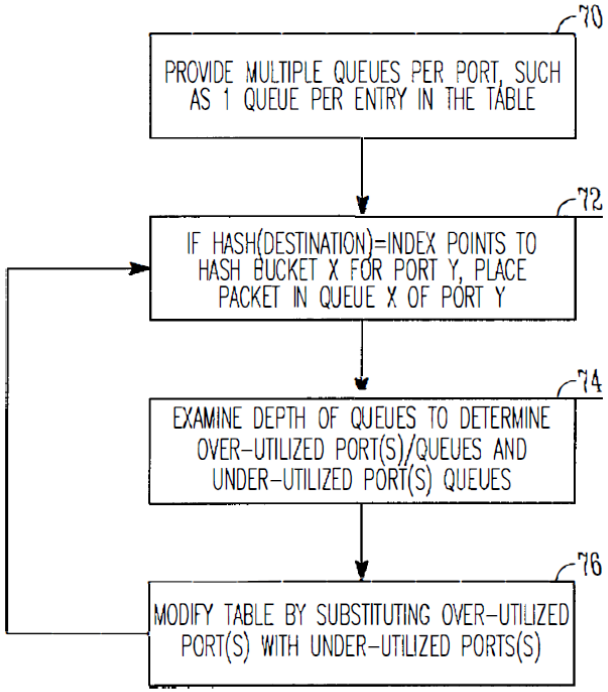
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

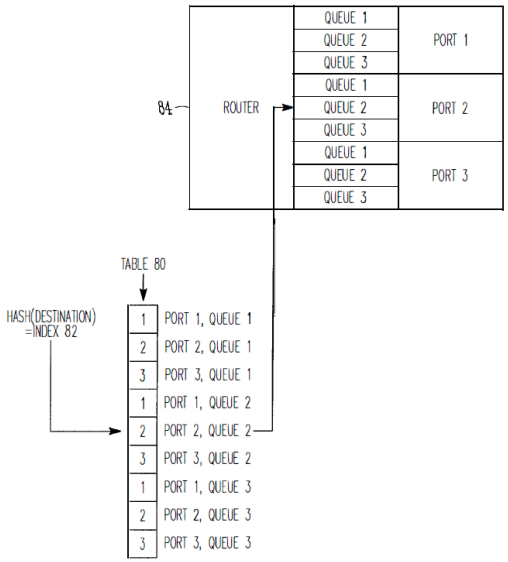
| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li ’914 at Figure 7 |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of |

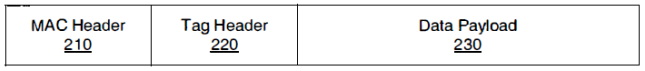
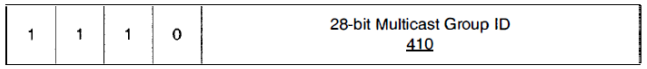
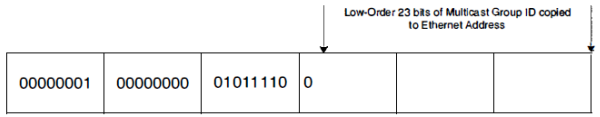
| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | | Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.”)<br><br>Li ’914 at 2:39-55 (“In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.”)<br><br>Li ’914 at 4:41-67 (“In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an |

| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | | index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.”)<br><br>Li ’914 at 5:13-28 (“In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.”)<br><br>Li ’914 at 5:29-41 (“Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.”) |

| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | | Li ’914 at 5:66-6:8 (“Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.”)<br><br>Li ’914 at 6:65-7:7 (“At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.”)<br><br>Borgione ’125 at 2:8-18 (“Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.”)<br><br>Borgione ’125 at 3:14-23 (“A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).”) |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | | Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast |

| No. | ’740 Patent Claim 9 | The Reference |
|---|---|---|
| | | packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.”)<br><br>Borgione ’125 at 5:28-54 (“The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.”)<br><br>Borgione ’125 at 7:1-5 (“A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).”)<br><br>Borgione ’125 at Figure 2-5 |

| No. | '740 Patent Claim 9 | The Reference |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| 10[a] | The method according to claim 9, wherein applying the hashing function comprises | The Reference discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | determining a hashing size responsively to a number of at least some of the first and second physical links, | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Singh, Solomon, and Alexander.<br><br>Below are examples of such references.<br><br>Singh at 15:15-39 ("The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars:<br><br># of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup )/32.<br><br>For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired.<br><br>Example to determine the number of arbiters and cross-bars for the following system:<br><br>4 channel port cards ( 40 Gbps)<br>8 links per channel<br>16 port cards<br>Speedup=l.5<br># of arbiters=( 4x2x2)/2=8 |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.")<br><br>Solomon at [0060] ("As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.")<br><br>Alexander at 3:1-40 ("The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest<br>values.<br><br>The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains $2^N$ entries occupying $2^N$ consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports.<br><br>Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.") |
| 10[b] | applying the hashing function to the at least one of the frame attributes to produce a hashing key, | The Reference discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, and Alexander.<br><br>Below is an example.<br><br>Alexander at 3:1-40 ("The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | values.<br><br>The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains $2^N$ entries occupying $2^N$ consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports.<br><br>Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.")<br><br>Alexander at 5:10-35 ("If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.")<br><br>Alexander at 5:36-46 ("The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.")<br><br>Alexander at 6:49-65 ("If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).") |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | |
| 10[c] | calculating a modulo of a division operation of the hashing key by the hashing size, and | The Reference discloses calculating a modulo of a division operation of the hashing key by the hashing size.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Singh, and Alexander.<br><br>Below are examples of such references.<br><br>Singh at 9:30-43 ("The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.")<br><br>Singh at 11:29-38 ("FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 15:15-39 ("The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars:<br><br># of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup )/32.<br><br>For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired.<br><br>Example to determine the number of arbiters and cross-bars for the following system:<br><br>4 channel port cards ( 40 Gbps)<br>8 links per channel<br>16 port cards<br>Speedup=l.5<br># of arbiters=( 4x2x2)/2=8<br># of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.")<br><br>Singh at 16:28-44 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.")<br><br>Singh at 17:31-49 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.")<br><br>Alexander at 5:36-46 ("The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.") |
| 10[d] | selecting the first and second physical links | The Reference discloses selecting the first and second physical links responsively to the modulo. |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | responsively to the modulo. | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Singh, and Alexander.<br><br>Below are examples of such references.<br><br>Singh at 9:30-43 ("The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.")<br><br>Singh at 11:29-38 ("FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.") |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | Singh at 15:15-39 ("The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars:<br><br># of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup )/32.<br><br>For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired.<br><br>Example to determine the number of arbiters and cross-bars for the following system:<br><br>4 channel port cards ( 40 Gbps)<br>8 links per channel<br>16 port cards<br>Speedup=l.5<br># of arbiters=( 4x2x2)/2=8<br># of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.")<br><br>Singh at 16:28-44 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo |

| No. | '740 Patent Claim 10 | The Reference |
|---|---|---|
| | | 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.")<br><br>Singh at 17:31-49 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.")<br><br>Alexander at 5:36-46 ("The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.") |

| No. | '740 Patent Claim 11 | The Reference |
|---|---|---|
| 11 | The method according to claim 10, wherein selecting | The Reference discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second |

| No. | '740 Patent Claim 11 | The Reference |
|---|---|---|
| | the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. | physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Solomon, and Alexander.<br><br>Below are examples of such references.<br><br>Solomon at [0054] ("Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.")<br><br>Solomon at [0056] ("Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which |

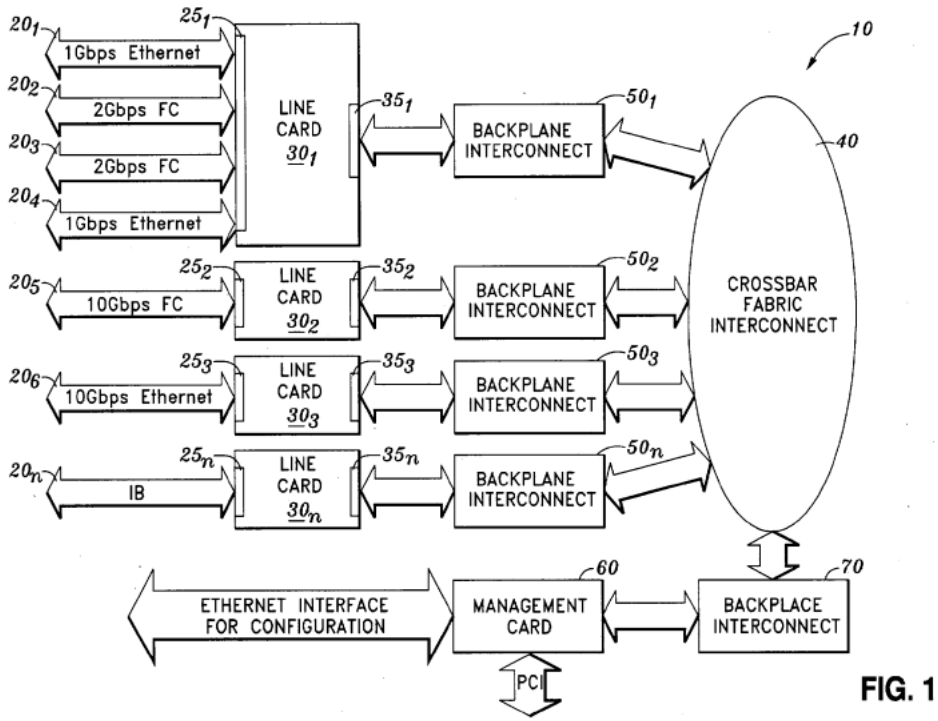
| No. | '740 Patent Claim 11 | The Reference |
|---|---|---|
| | | was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.")<br><br>Alexander at 5:36-46 ("The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.") |

| No. | '740 Patent Claim 12 | The Reference |
|---|---|---|
| 12 | The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. | The Reference discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

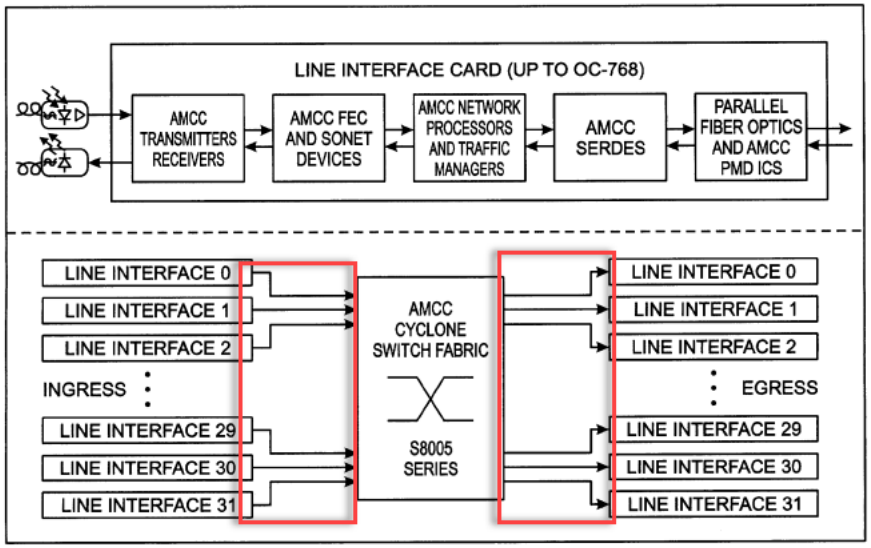
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| 13[preamble] | A method for communication, comprising: | The Reference discloses a method for communication.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

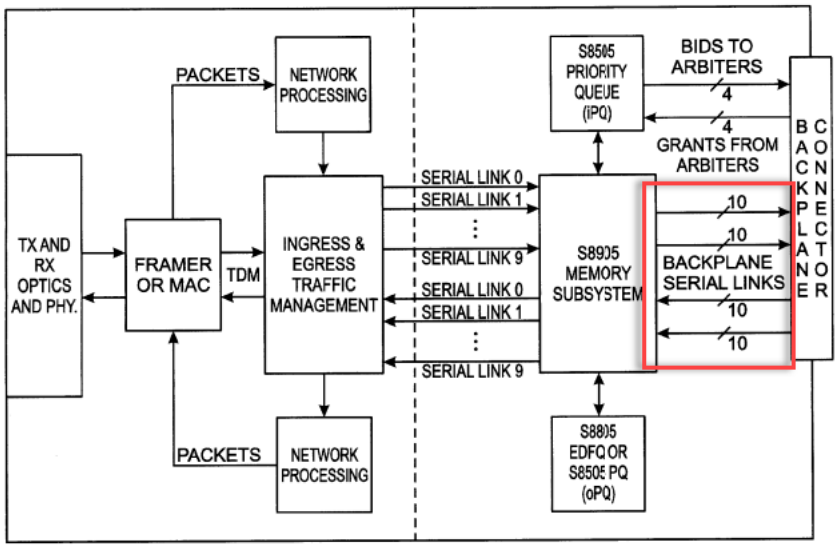
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 13[a] | coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel; | The Reference discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 13[b] | coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel; | The Reference discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified |

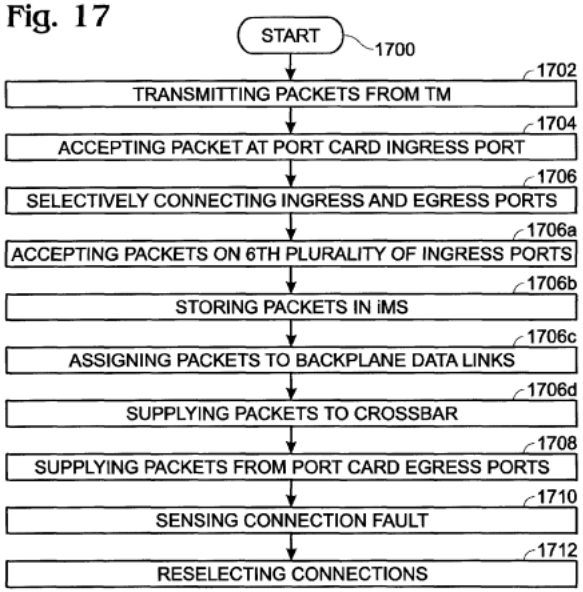
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

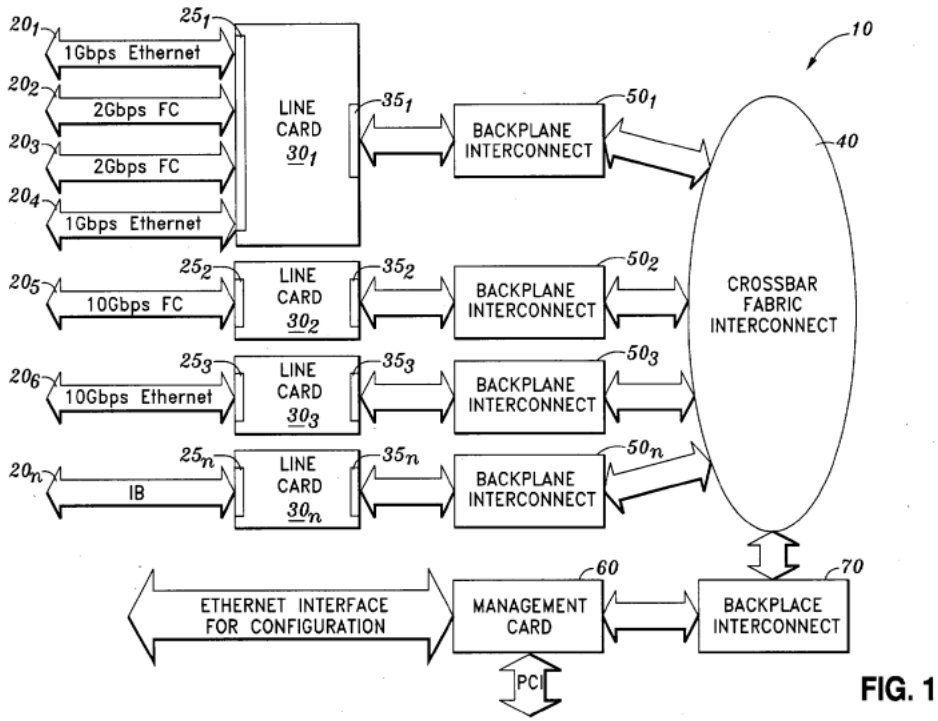
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

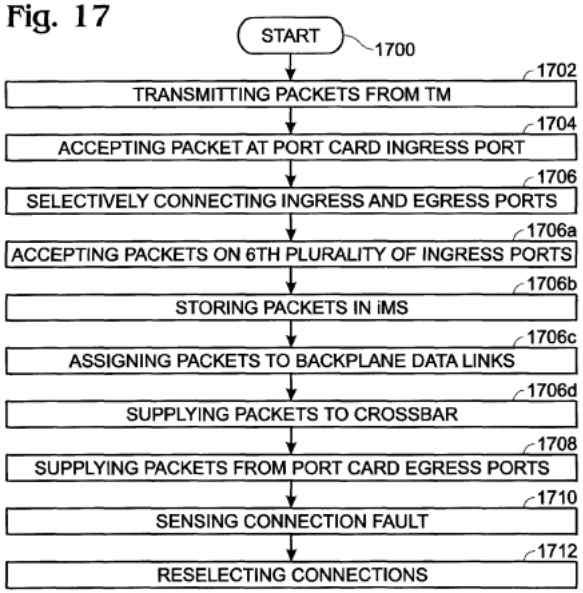
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 13[c] | receiving a data frame having frame attributes sent between the communication network and the network node: | The Reference discloses receiving a data frame having frame attributes sent between the communication network and the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 13[d] | selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical | The Reference discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

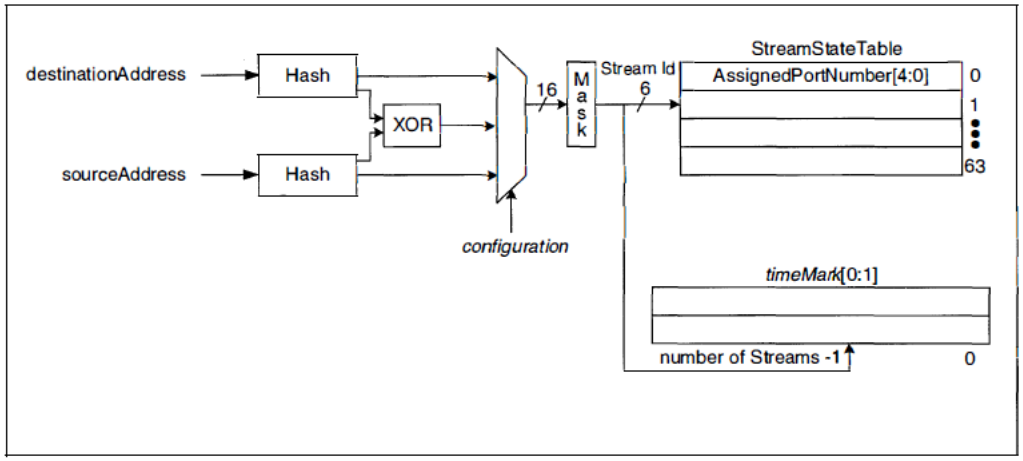
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | link out of the second group; and | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

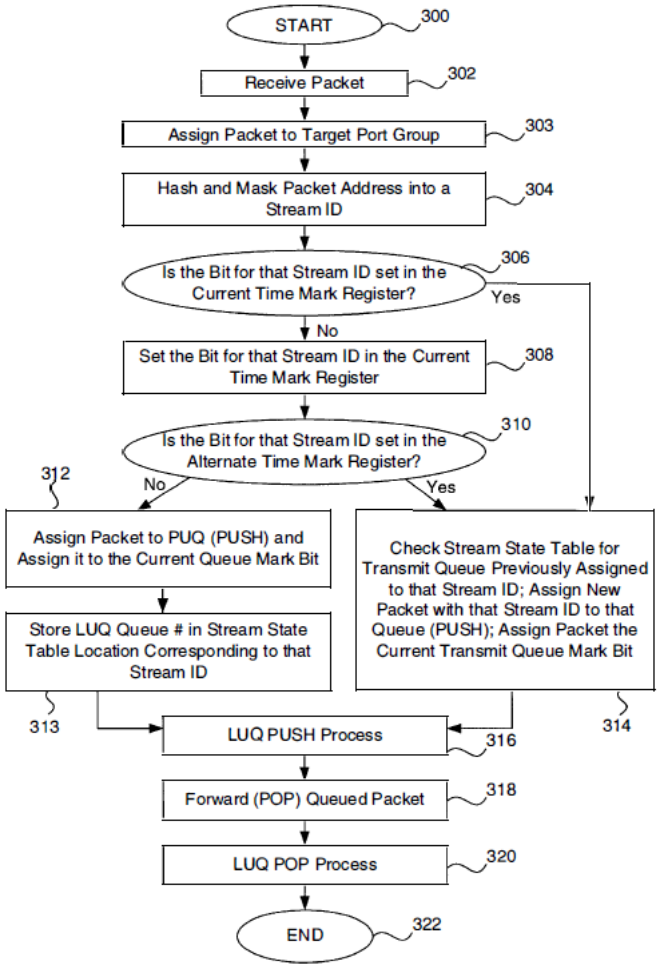
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including selecting physical links over which to send a packet. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | • Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

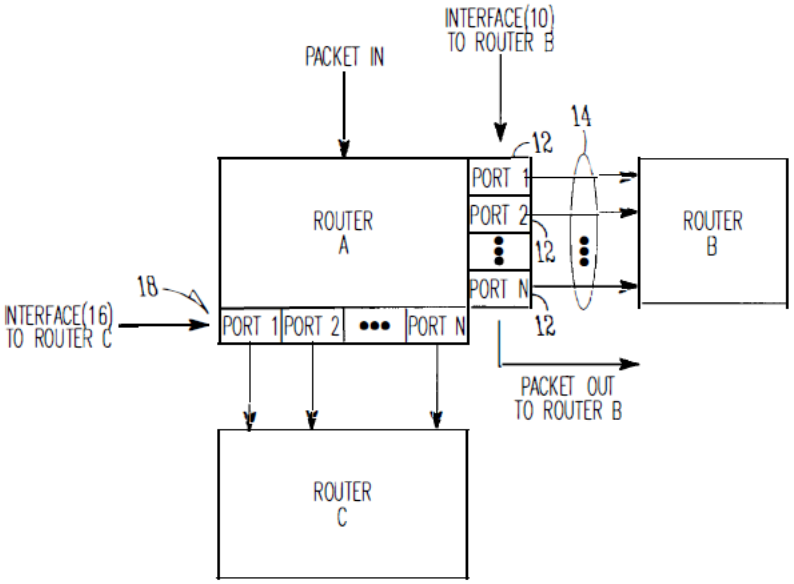
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

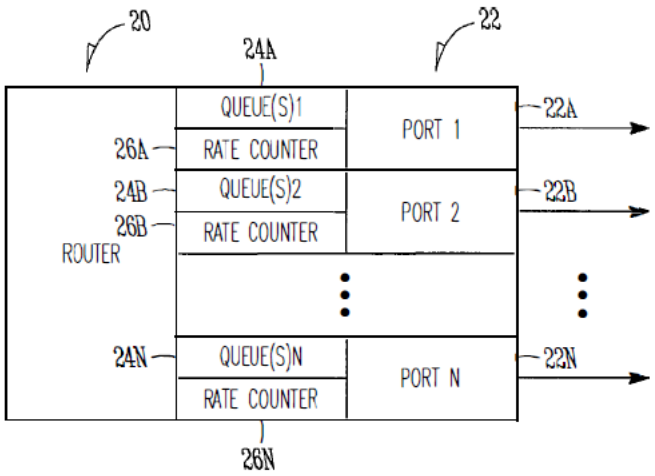
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

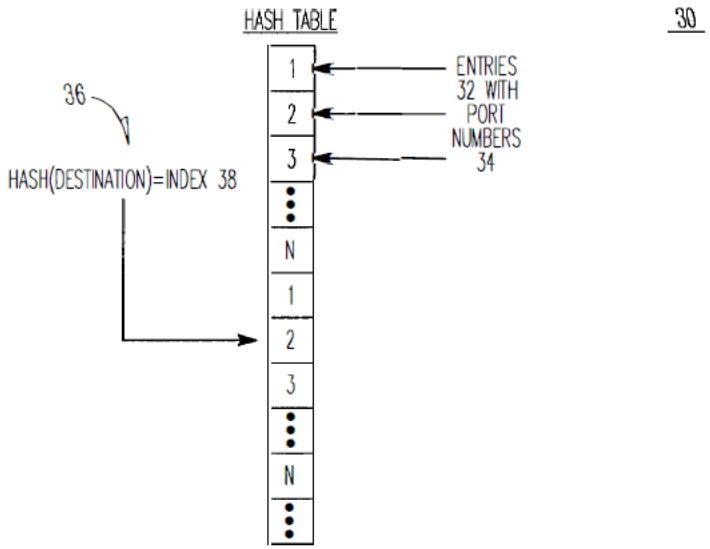
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

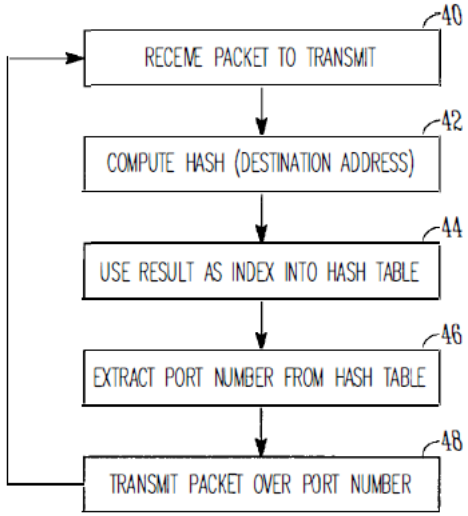
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

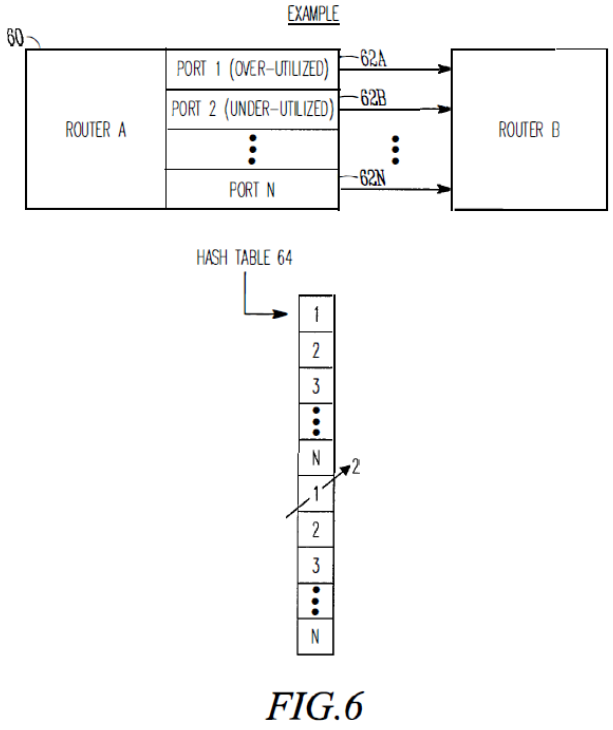
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

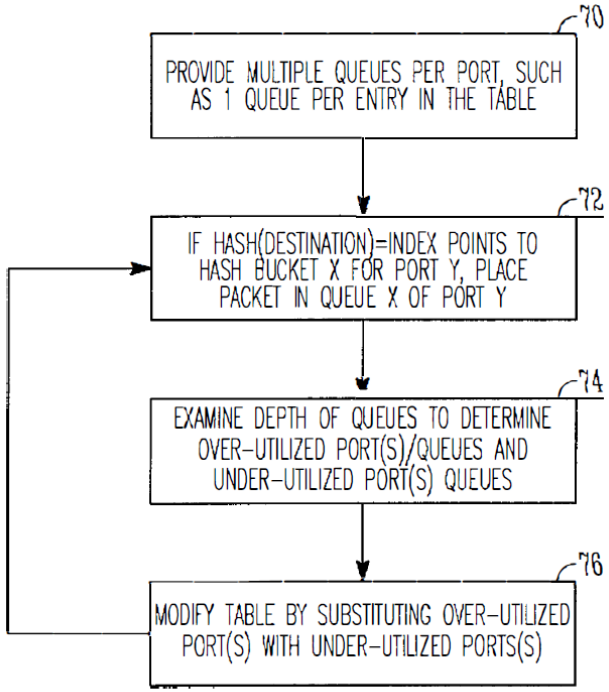
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

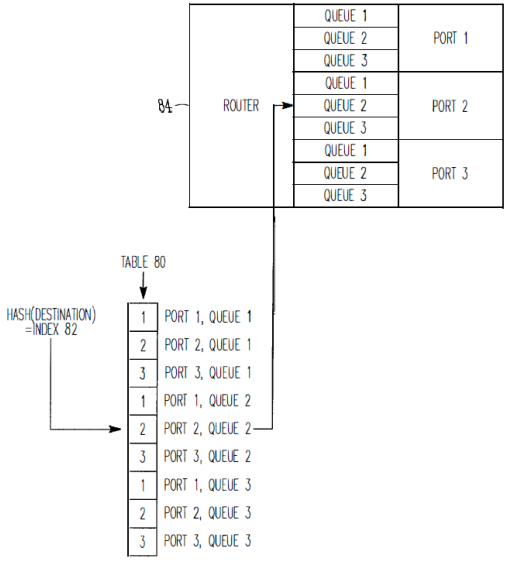
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br><br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

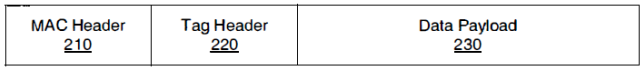
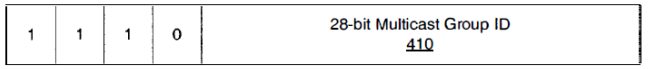
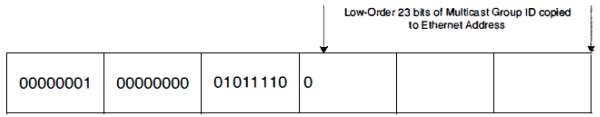
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

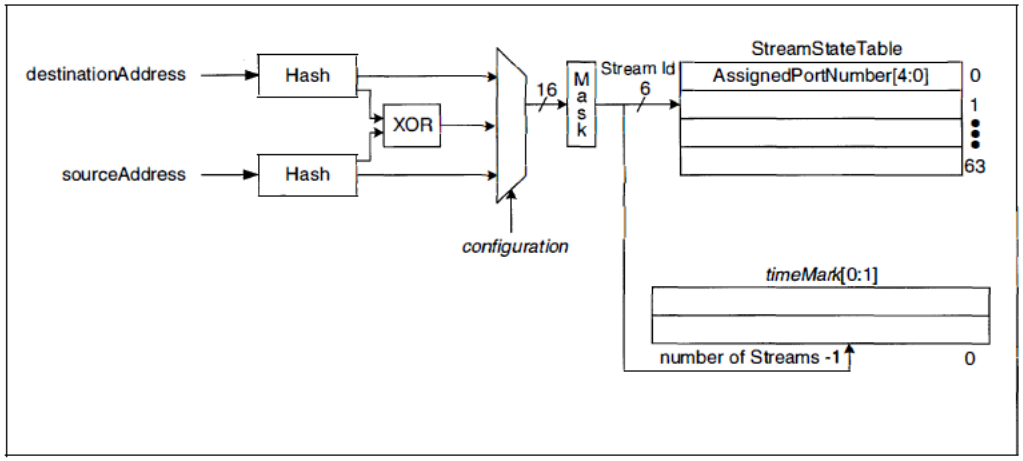
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

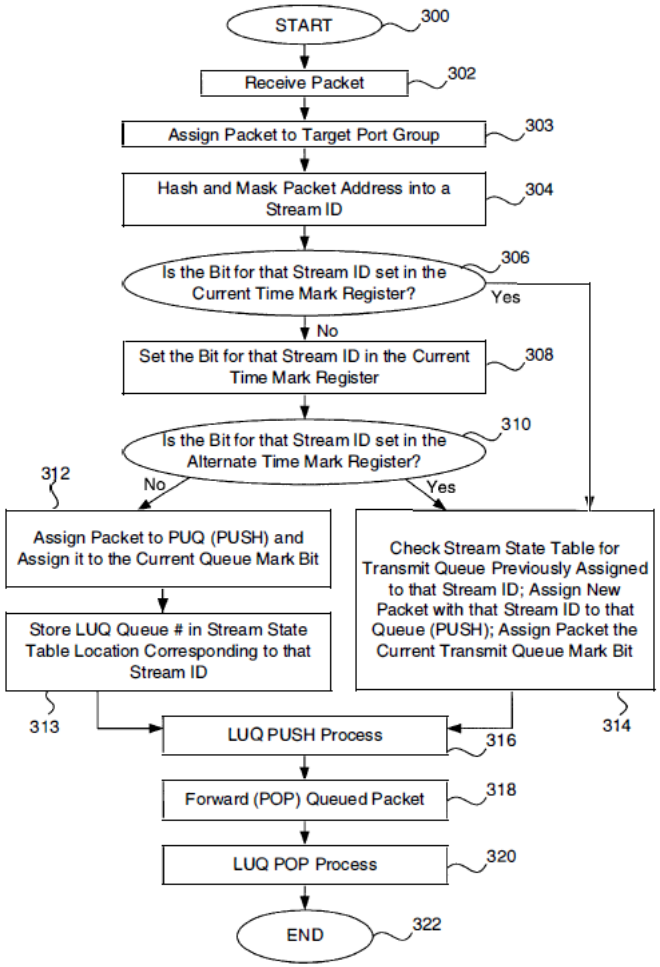
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| 13[e] | sending the data frame over the selected first and second physical links, | The Reference discloses sending the data frame over the selected first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

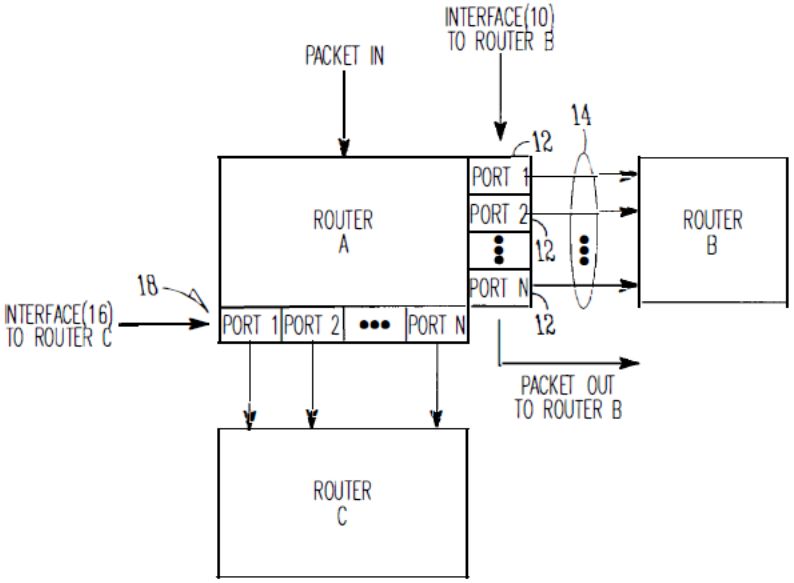
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

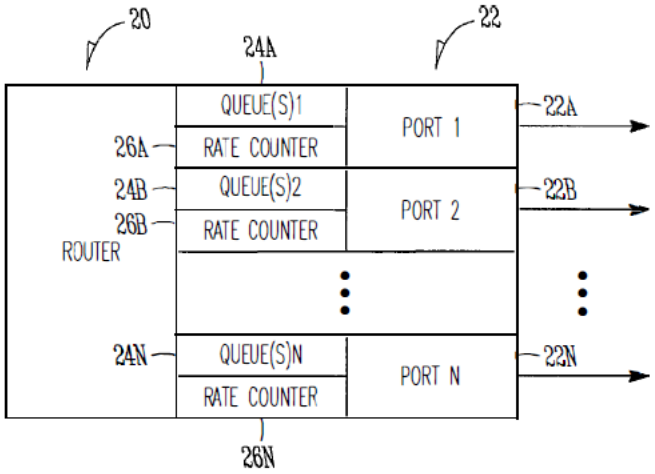
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

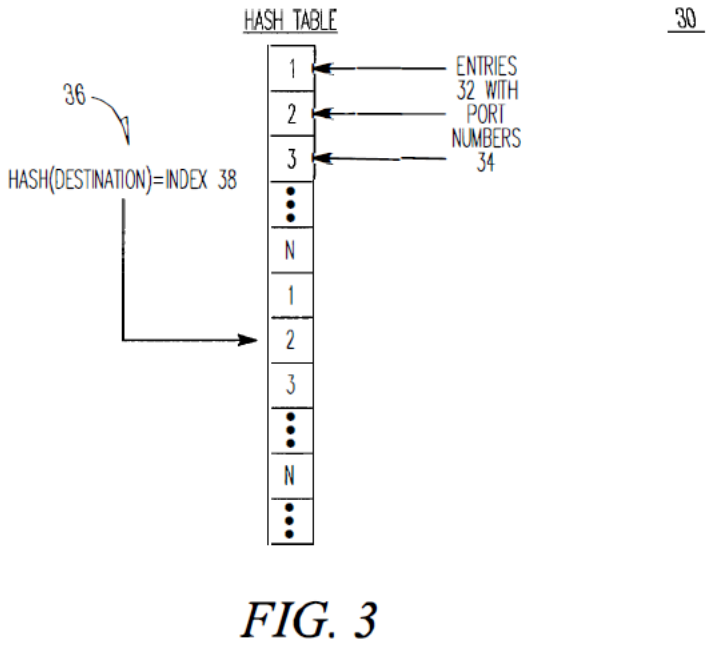
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

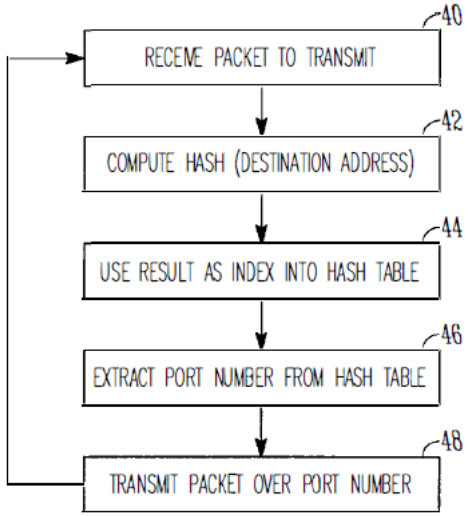
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

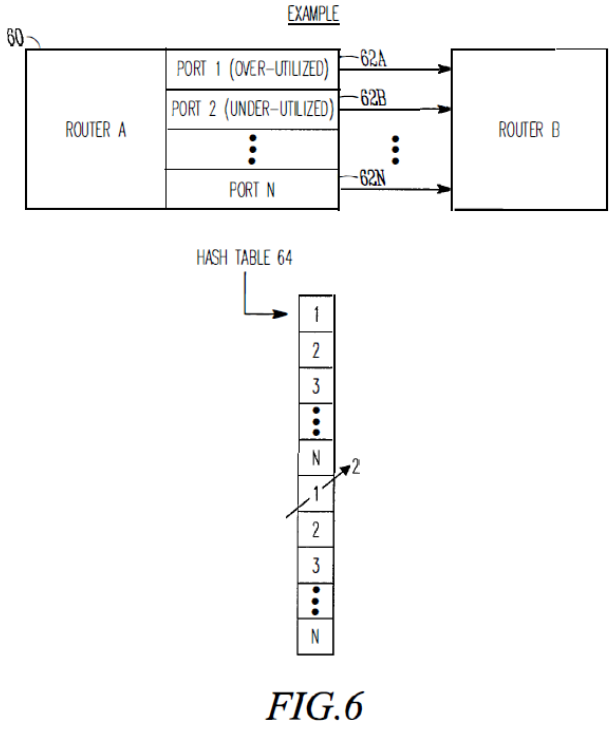
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

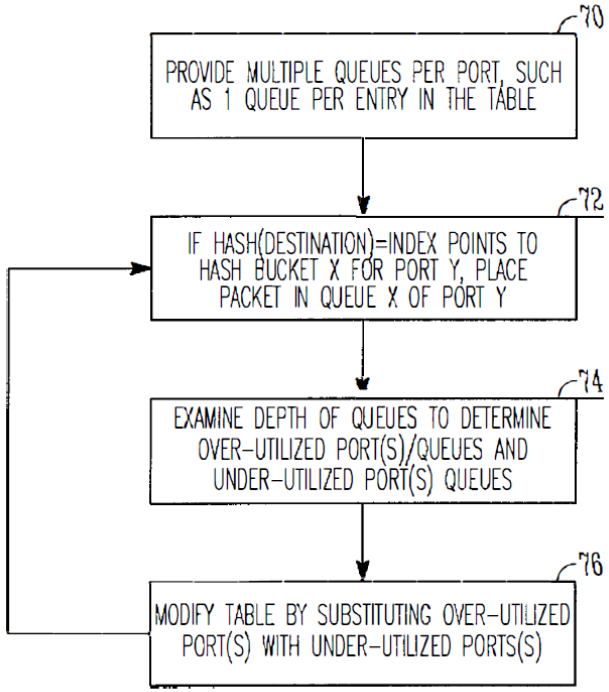
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

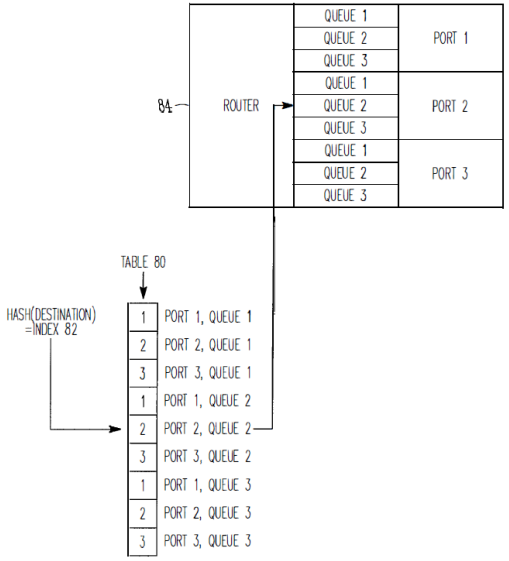
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

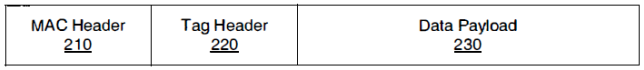
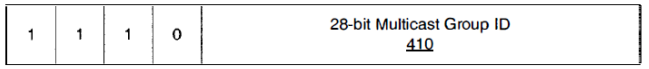
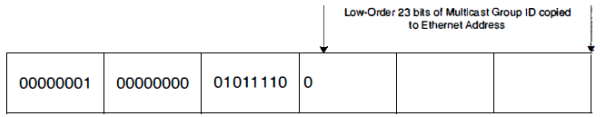
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

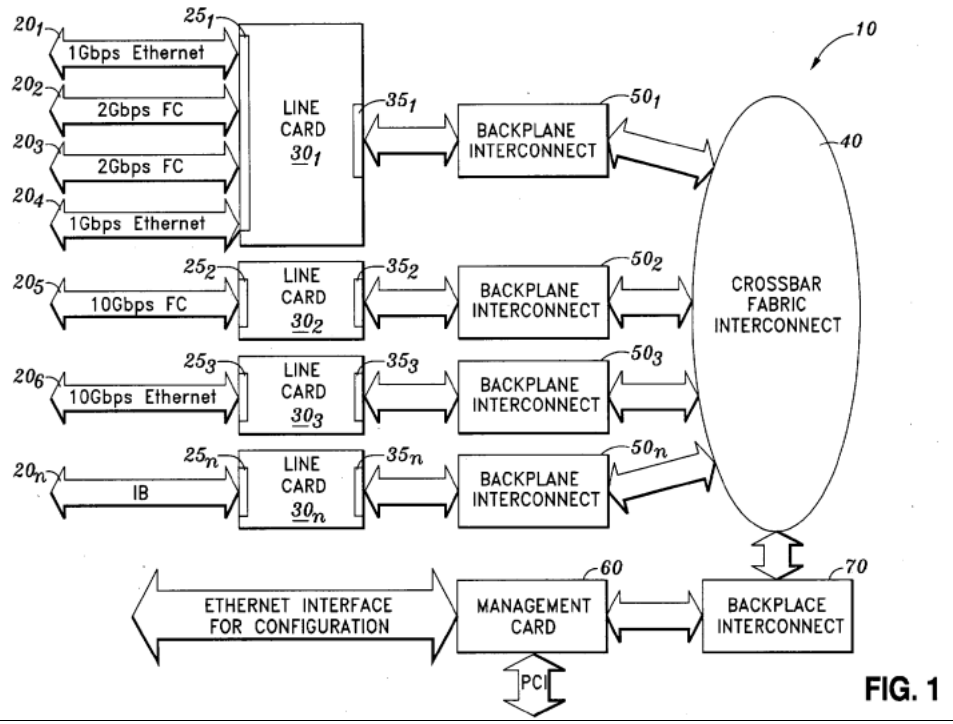
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| 13[f] | coupling the network node to the one or more interface modules and | The Reference discloses coupling the network node to the one or more interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 13[g] | coupling each of the one or more interface modules to the communication network comprising | The Reference discloses coupling each of the one or more interface modules to the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 13[h] | specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess | The Reference discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| | information rate (EIR) of a communication service provided by the communication network to the network node, and | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, and Solomon.<br><br>Below is an example.<br><br>Solomon at [0023] ("In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.")<br><br>Solomon at [0050] ("The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 ( comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).") |

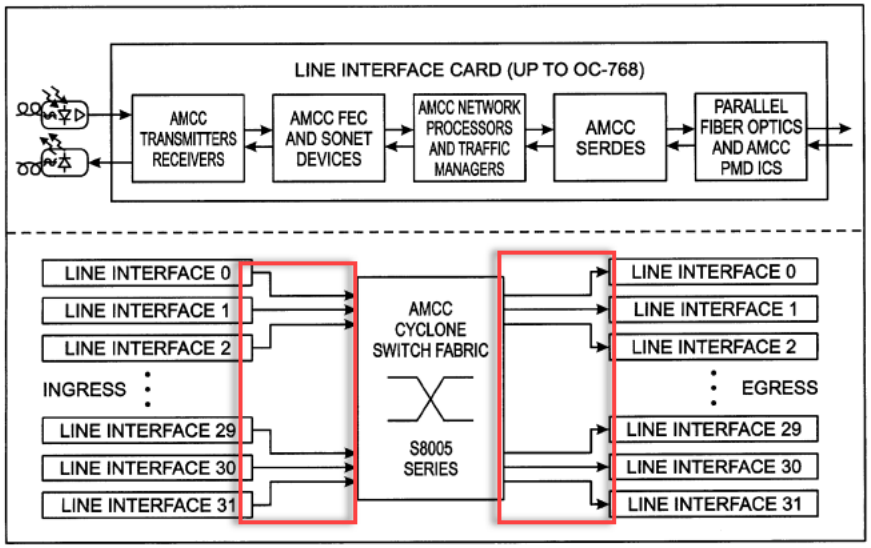
| No. | '740 Patent Claim 13 | The Reference |
|---|---|---|
| 13[i] | allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. | The Reference discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

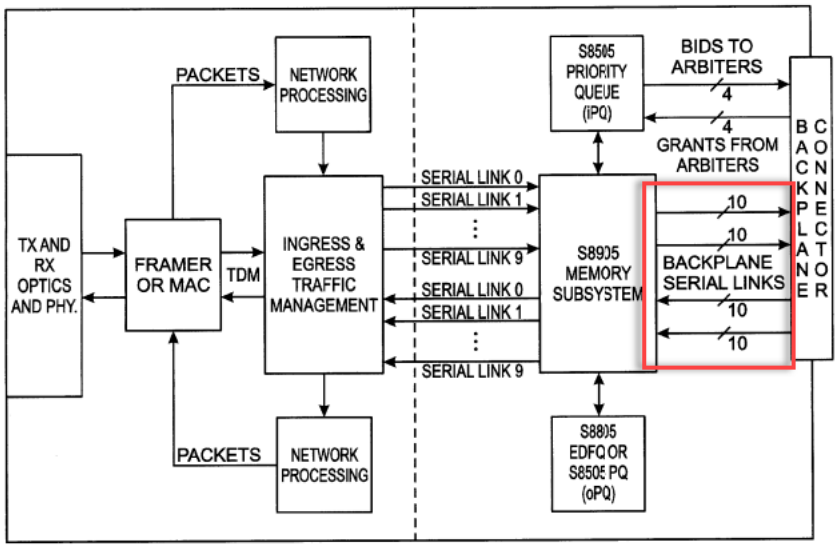
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| 14[preamble] | A method for connecting user ports to a communication network, comprising: | The Reference discloses a method for connecting user ports to a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 14[a] | coupling the user ports to one or more user interface modules; | The Reference discloses coupling the user ports to one or more user interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

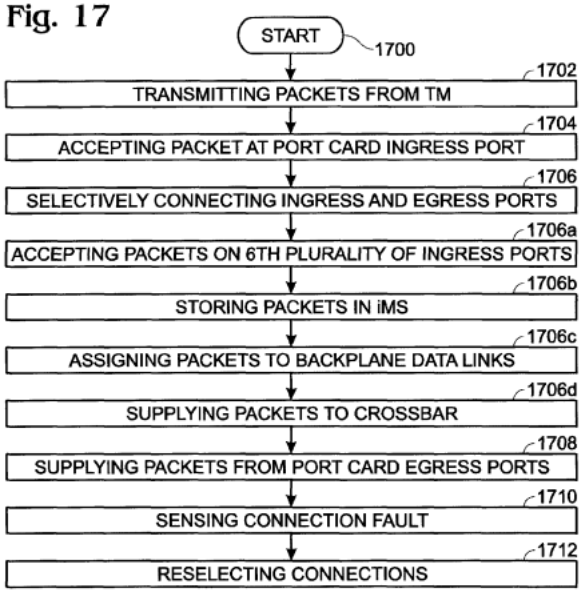
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 14[b] | coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel, | The Reference discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1<br>  <br><br>FIG. 1 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

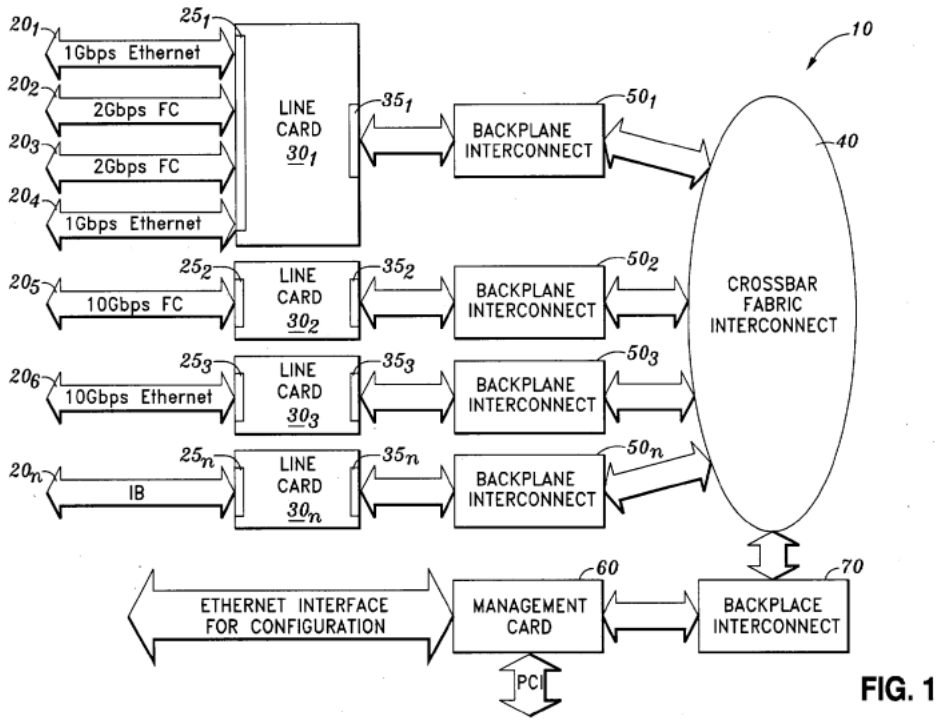
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

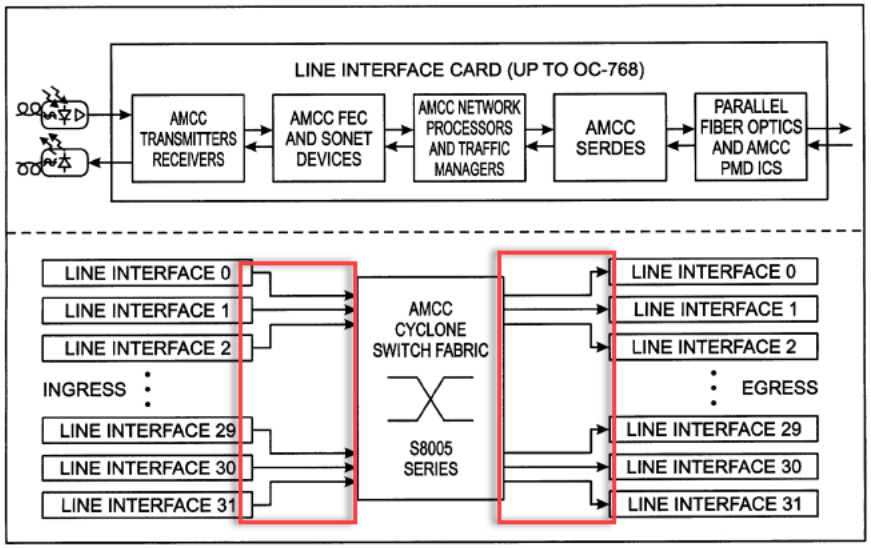
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

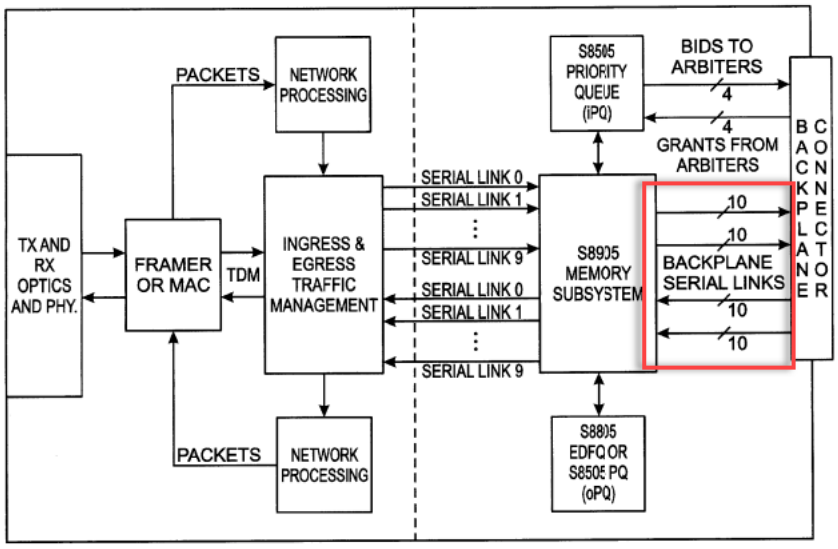
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 14[c] | at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction; | The Reference discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces |

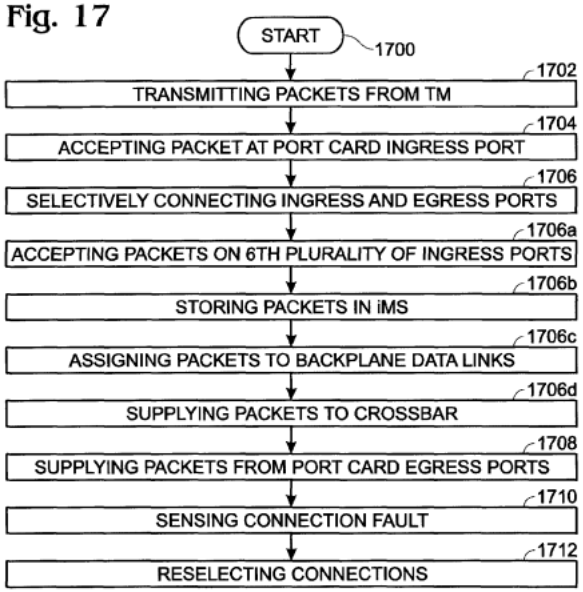
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

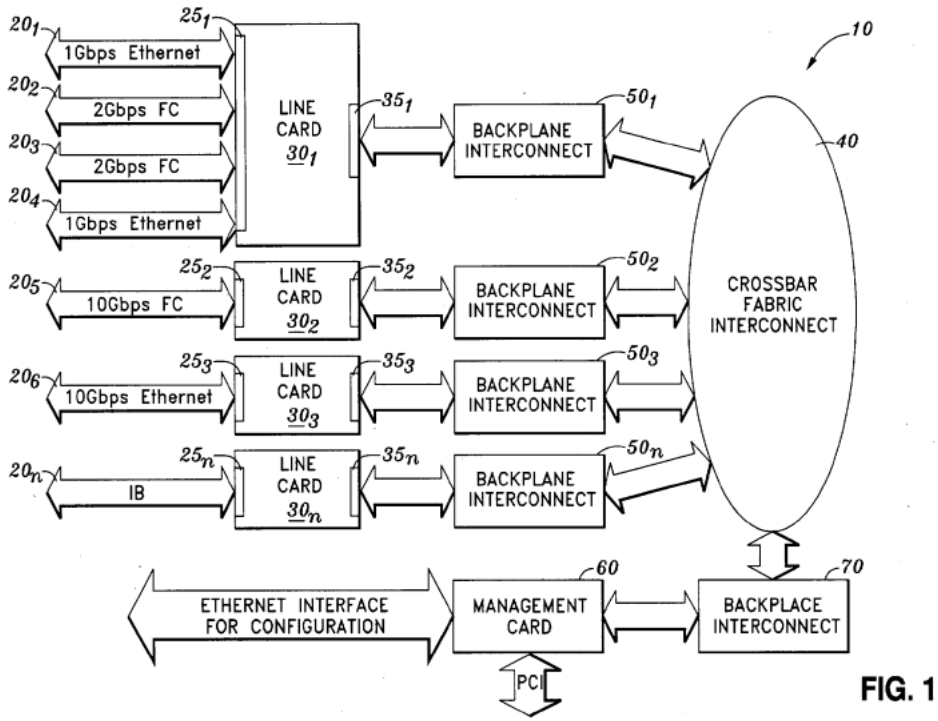
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

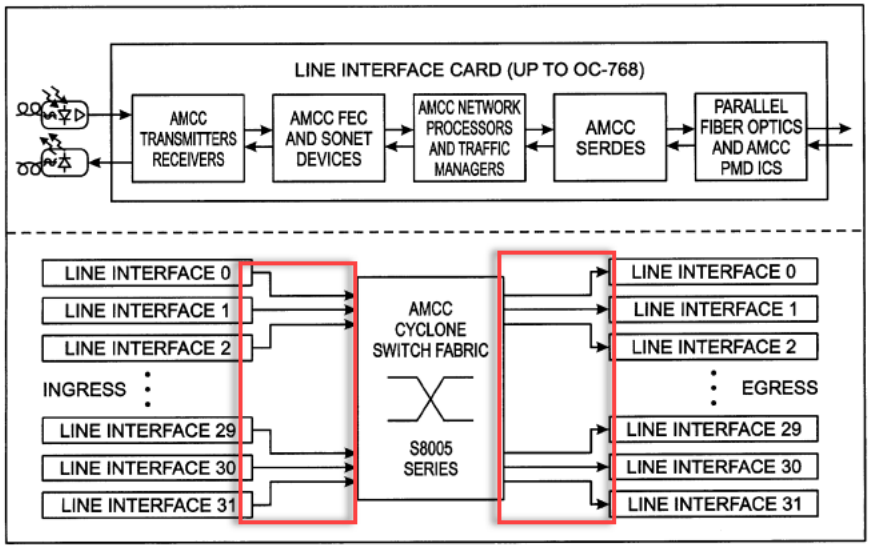
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

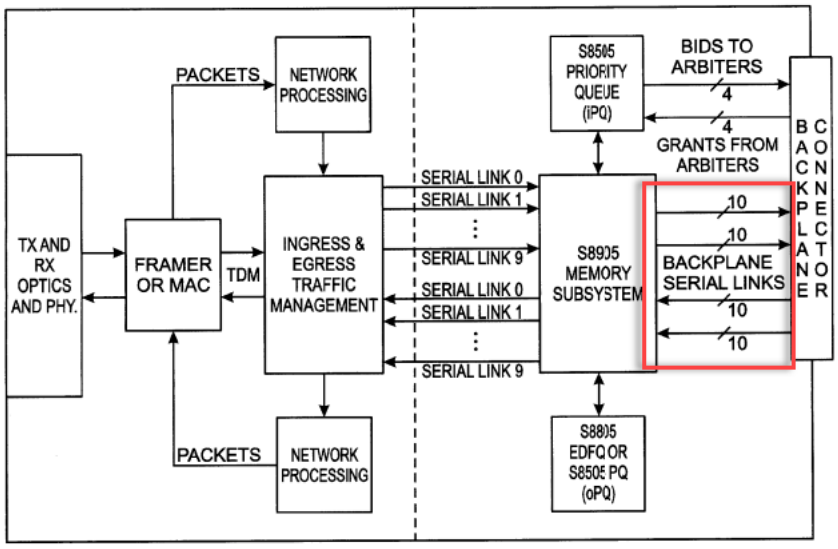
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 14[d] | receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes; | The Reference discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 14[e] | for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or | The Reference discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

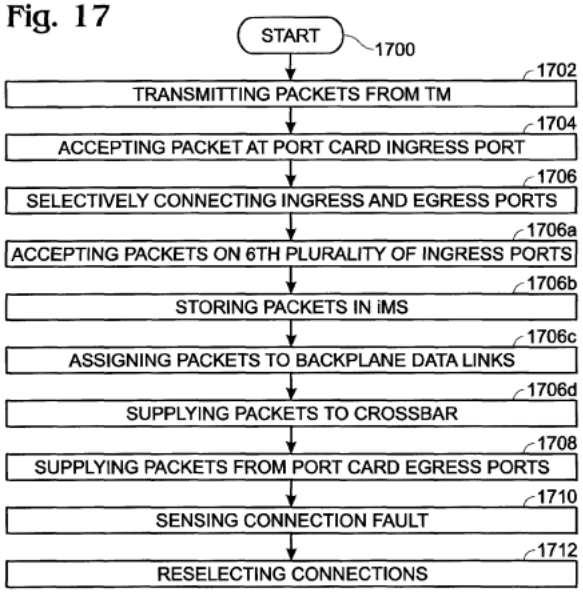
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | more backplane traces; and | skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

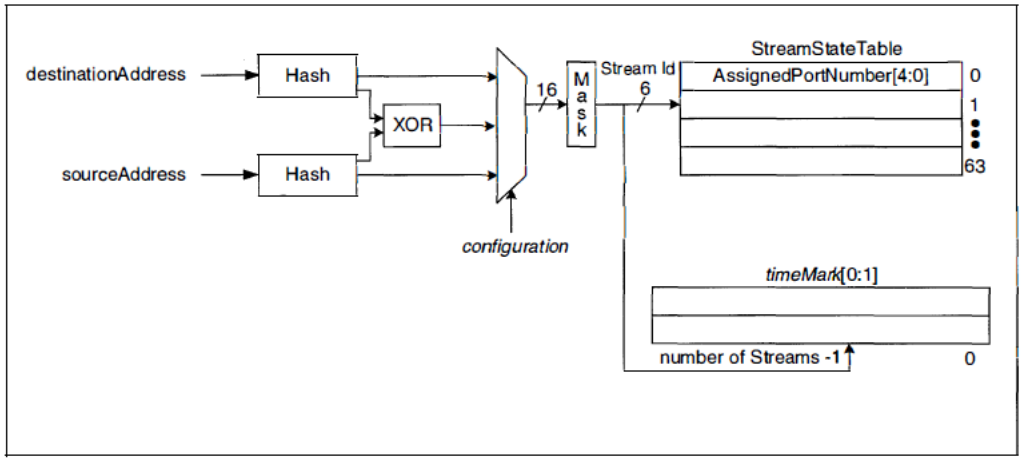
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

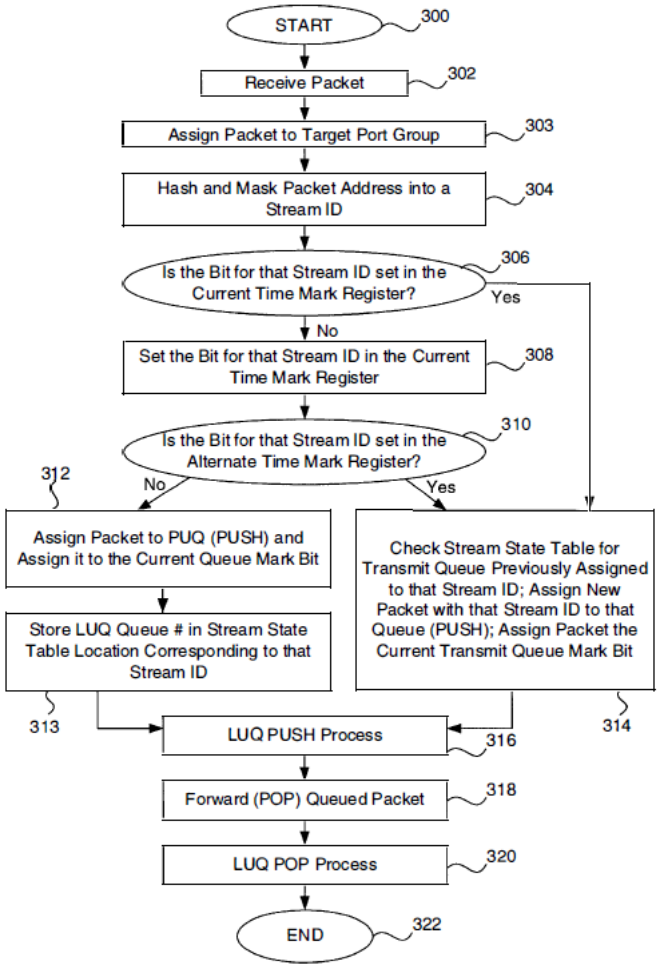
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

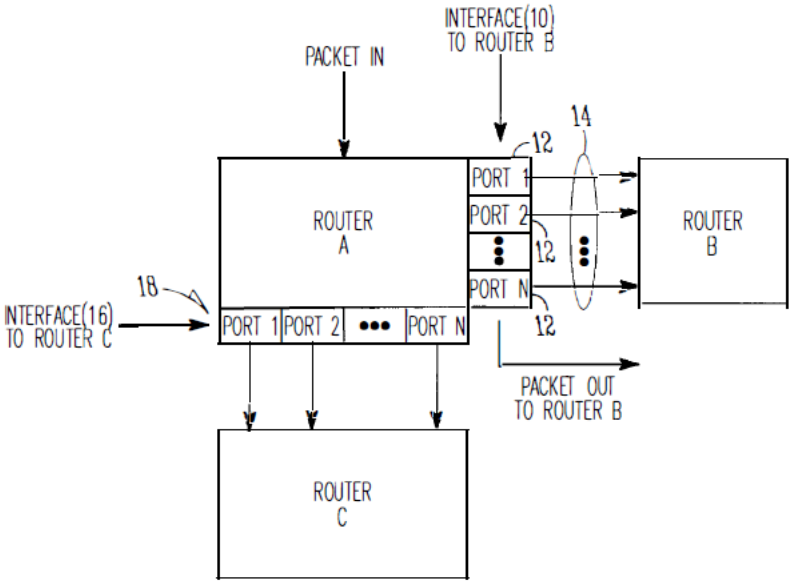
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

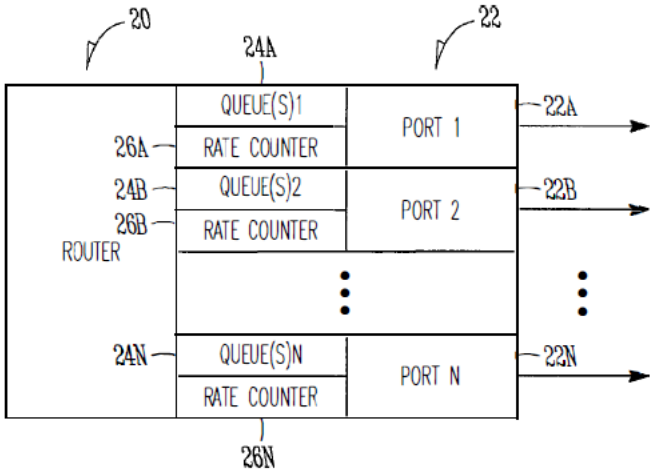
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

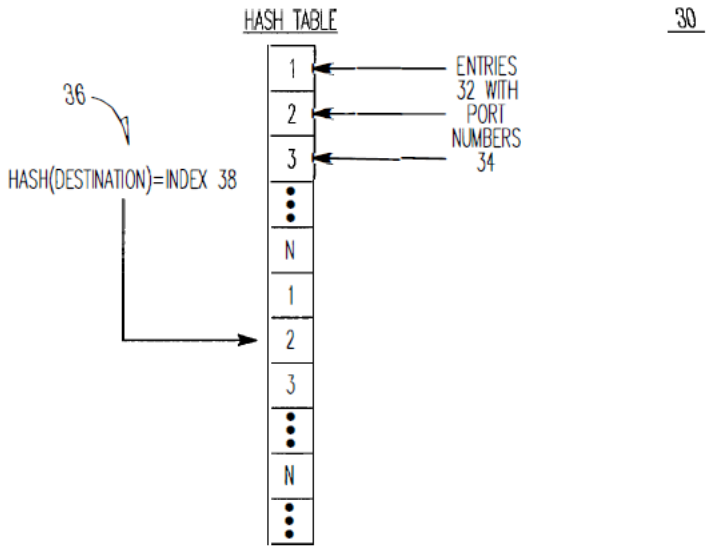
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

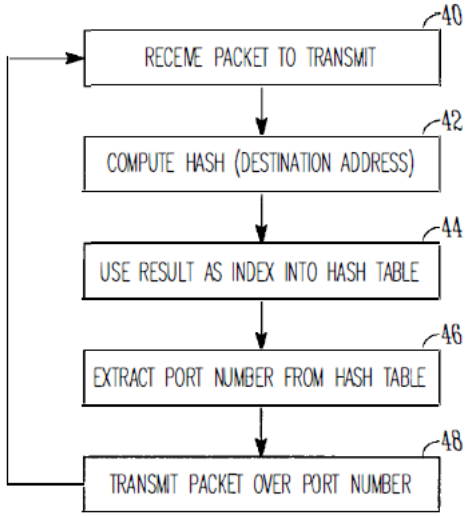
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

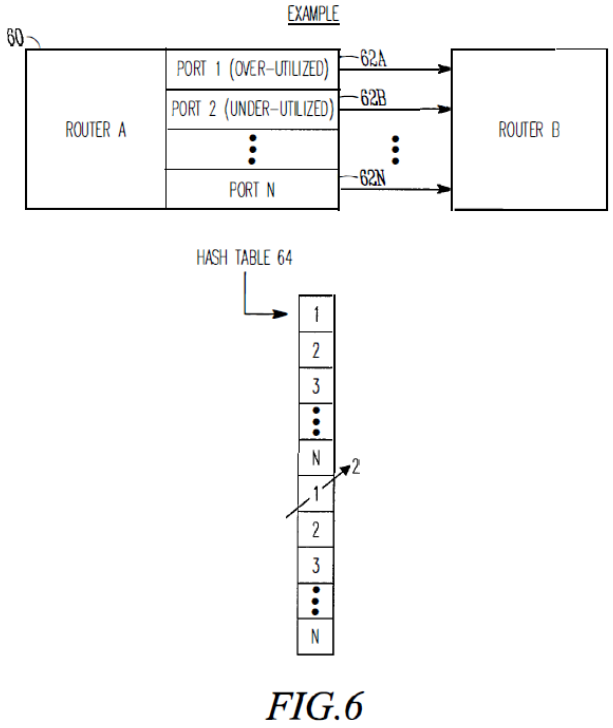
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

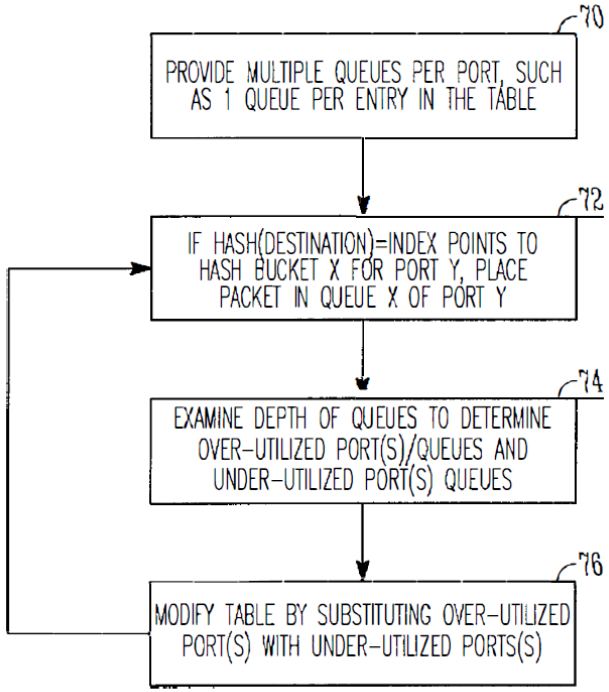
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

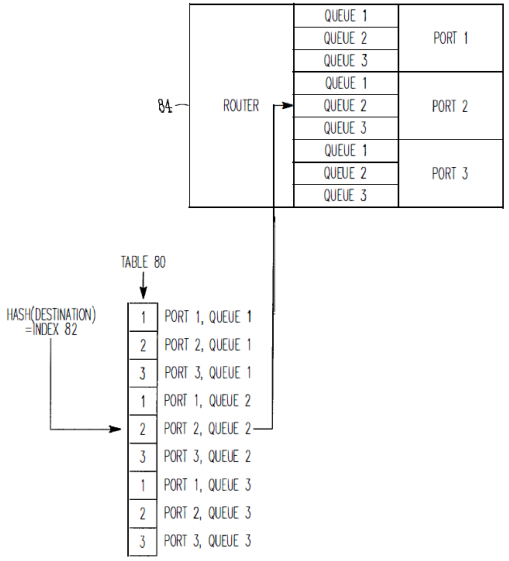
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

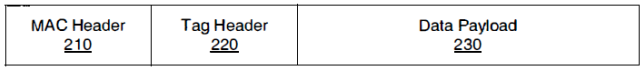
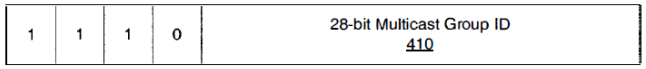
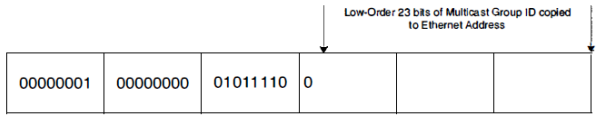
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

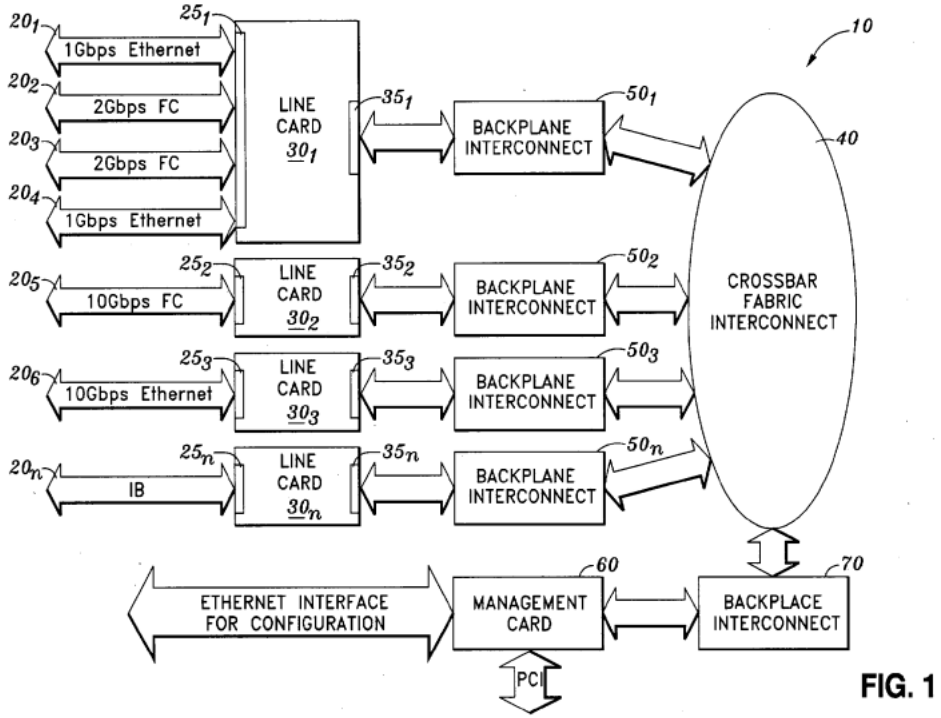
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

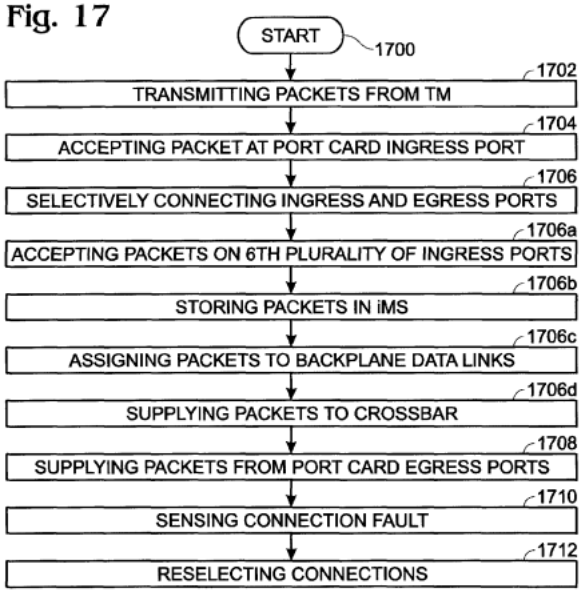
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br><br>  <br><br>**Figure 2**<br><br>  <br><br>**Figure 3**<br><br>  <br><br>**Figure 4**<br><br>  <br><br>**Figure 5** |

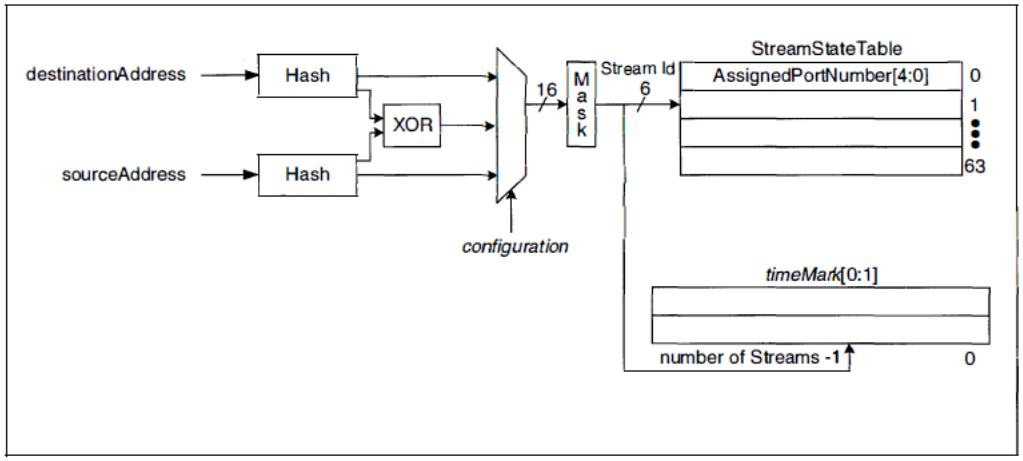
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| 14[f] | sending the data frame over the selected backplane trace; | The Reference discloses sending the data frame over the selected backplane trace.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may |

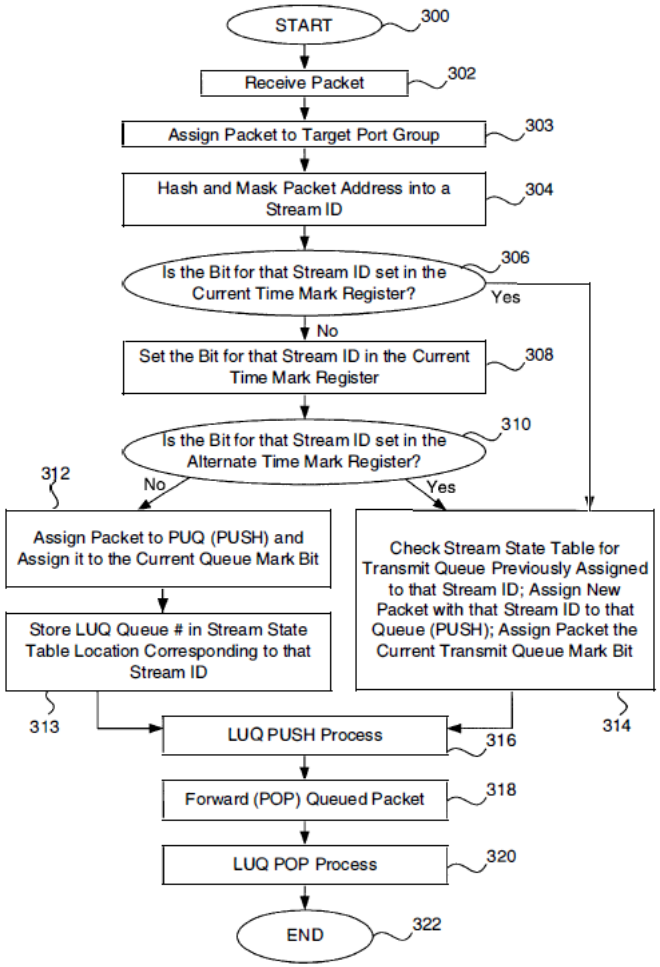
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1<br>  <br><br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).") |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

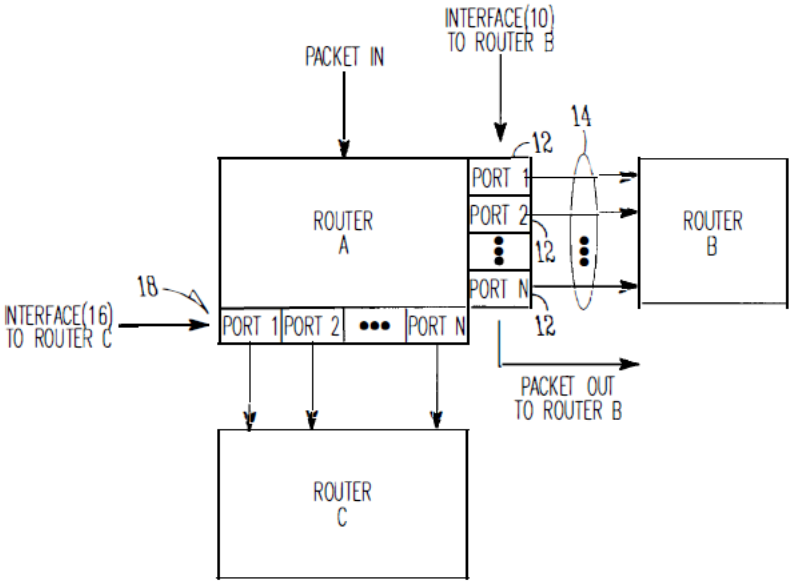
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

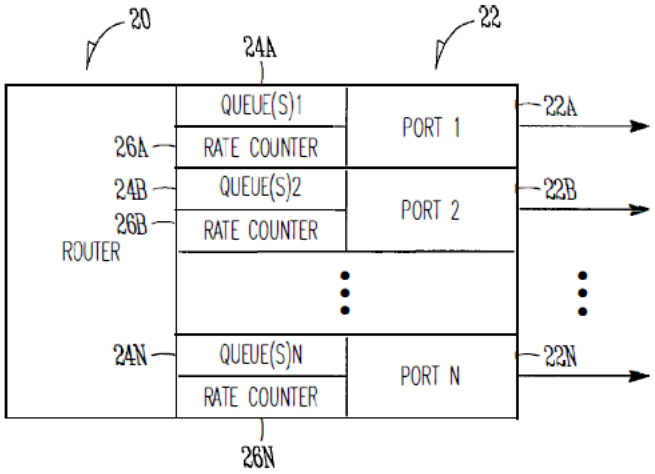
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

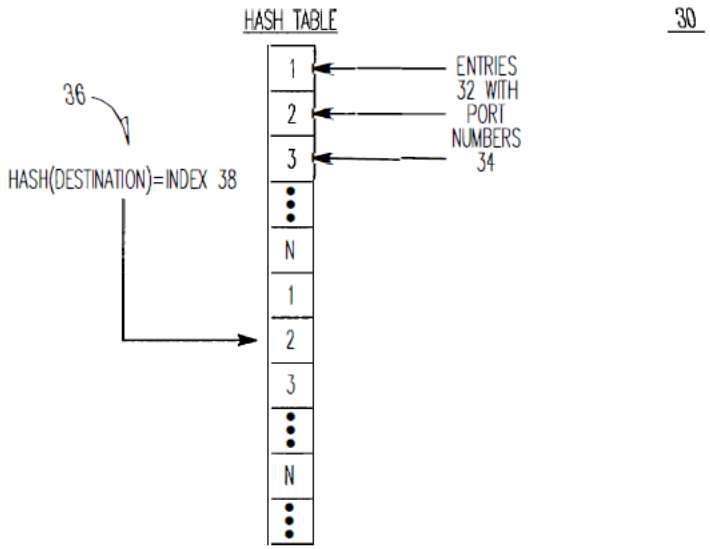
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

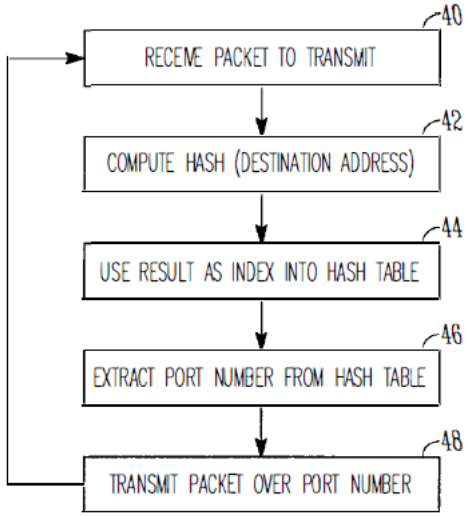
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

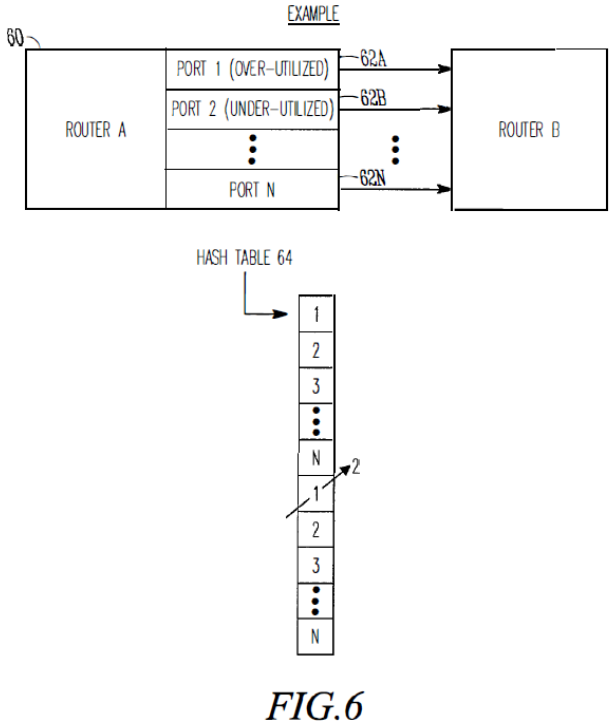
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

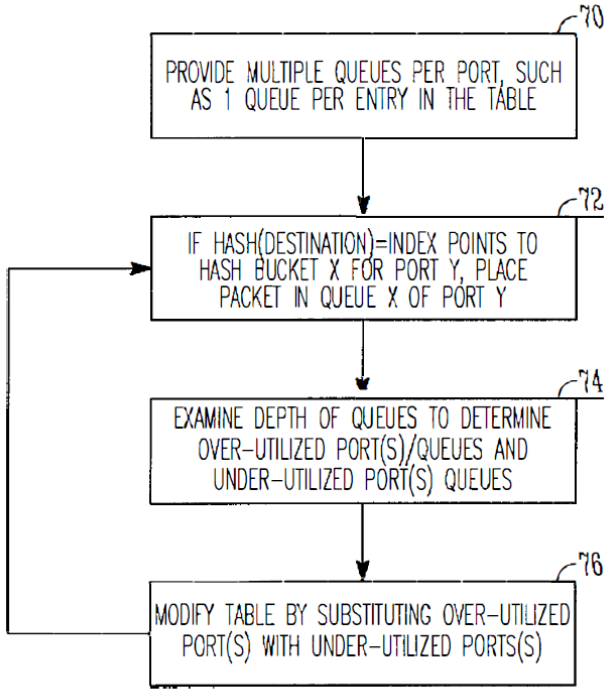
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

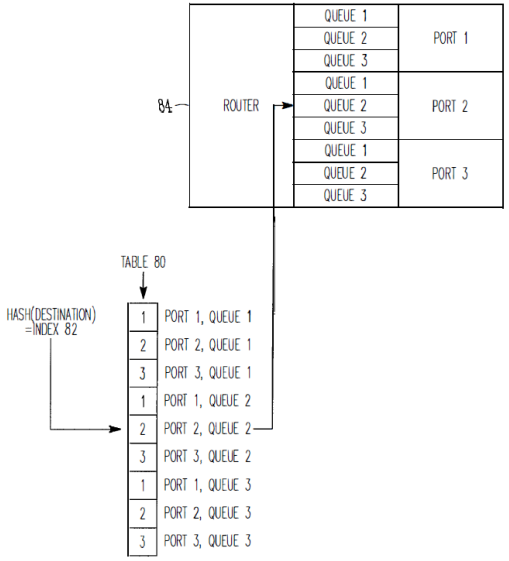
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

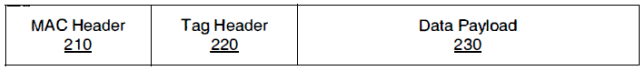
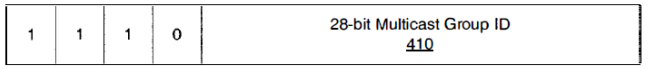
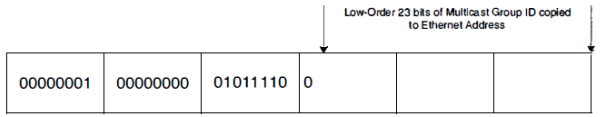
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

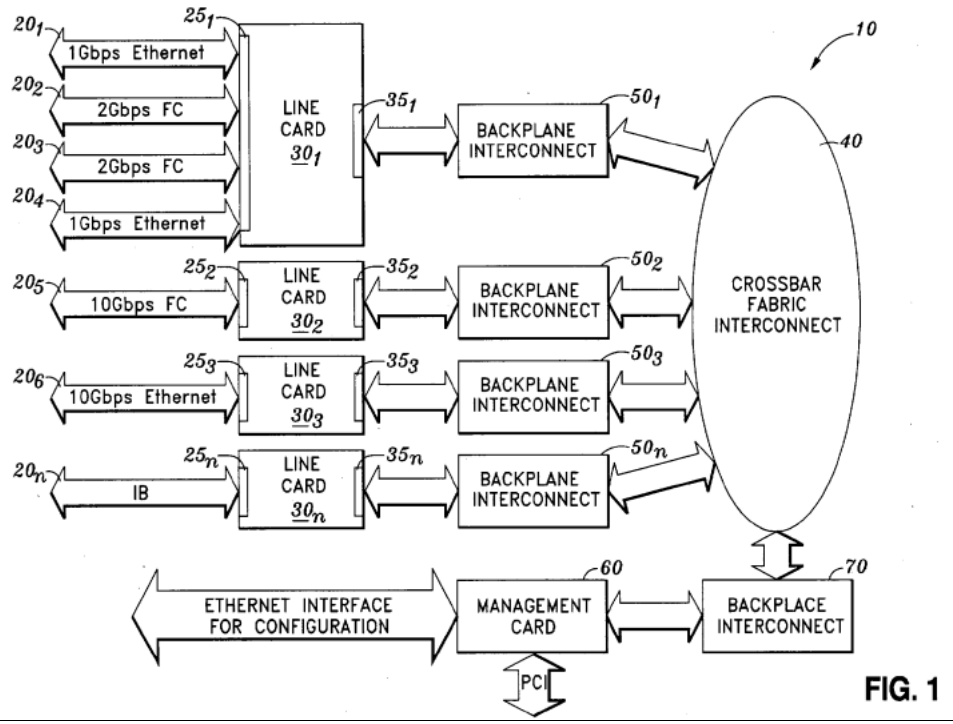
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

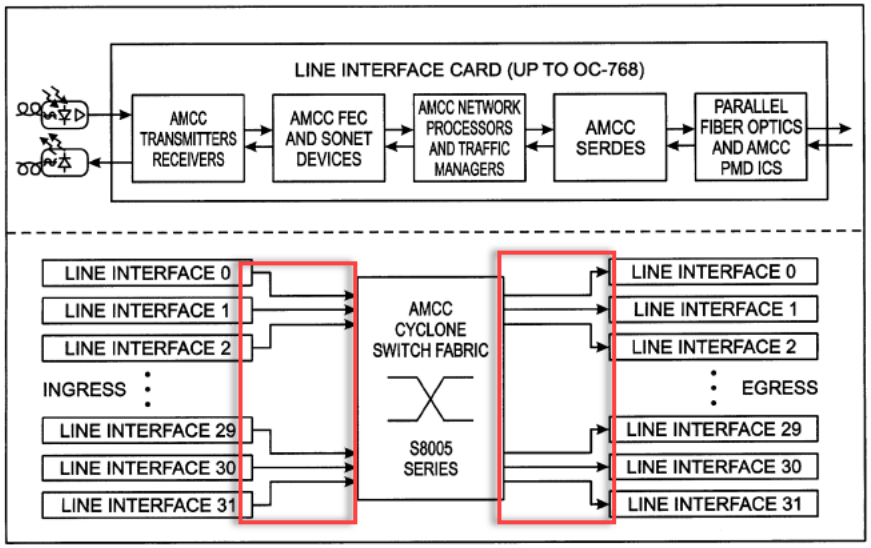
| No. | '740 Patent Claim 14 | The Reference |
|---|---|---|
| 14[g] | said sending comprising communicating along said at least one of said backplane traces. | The Reference discloses said sending comprising communicating along said at least one of said backplane traces.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

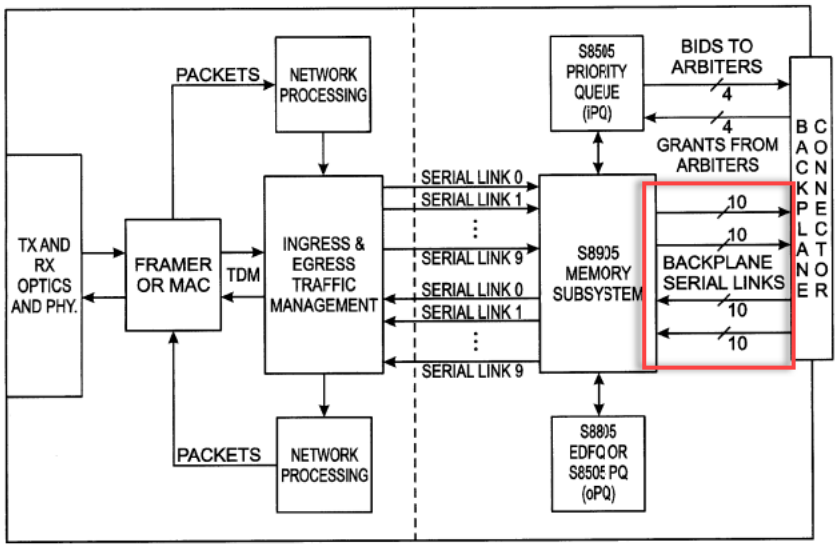
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| 15[preamble] | A method for connecting user ports to a communication network, comprising: | The Reference discloses a method for connecting user ports to a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 15[a] | coupling the user ports to one or more user interface modules; | The Reference discloses coupling the user ports to one or more user interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

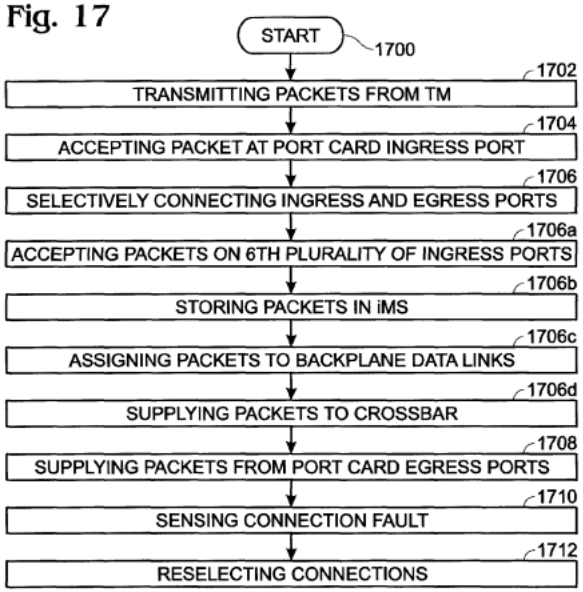
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 15[b] | coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel; | The Reference discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1<br>  <br><br>FIG. 1 |

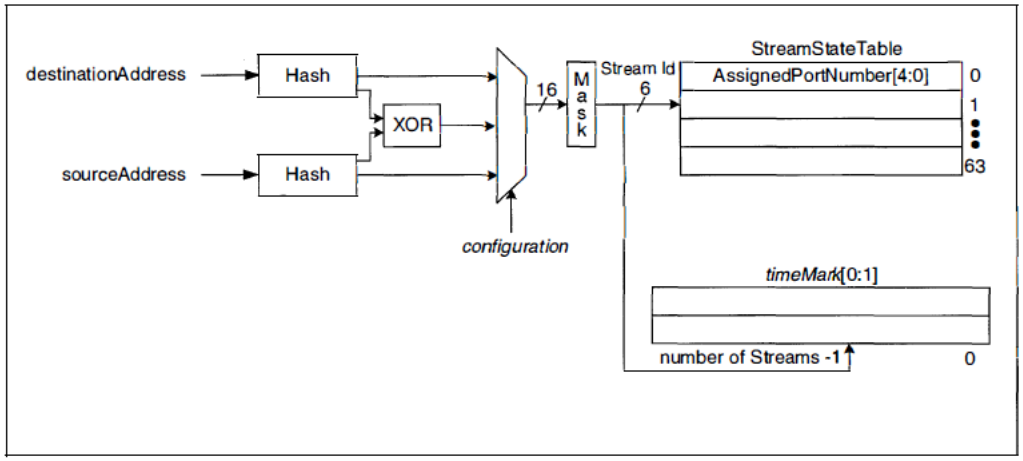
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

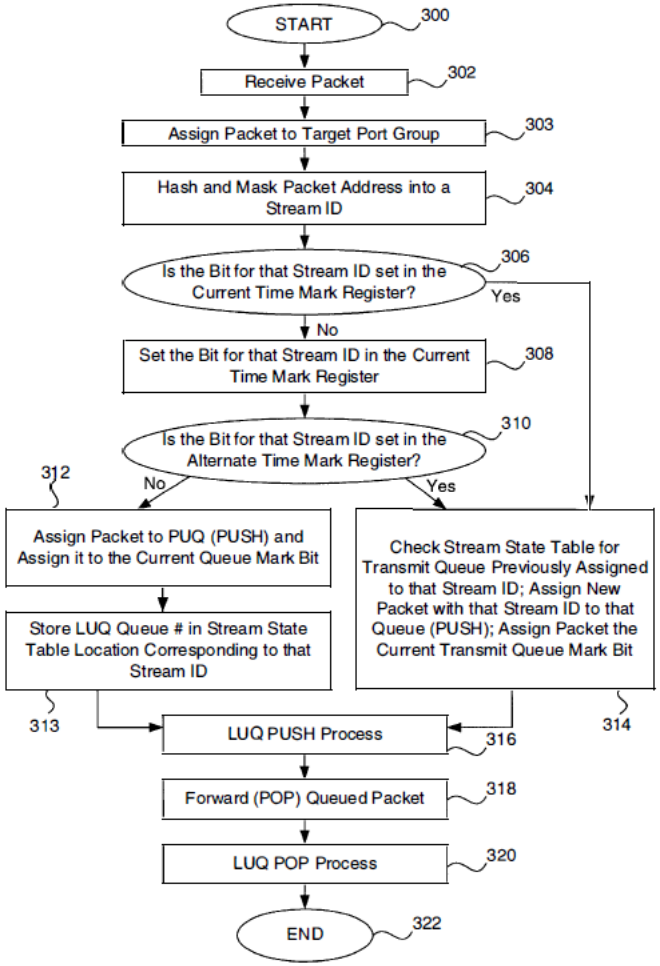
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 15[c] | receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes; | The Reference discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 15[d] | for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or | The Reference discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | more backplane traces; and | skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | DeJager '424 at Figure 3A<br>  <br><br>*FIG. 3A* |

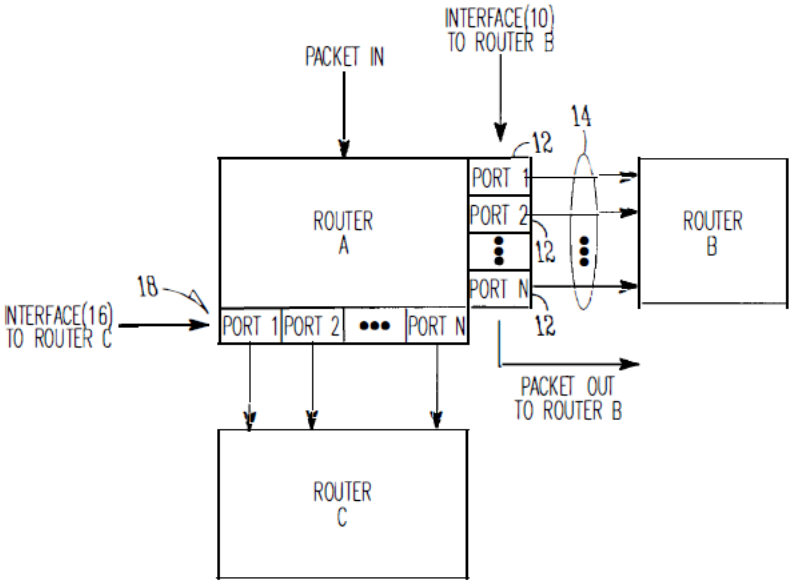
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.") |

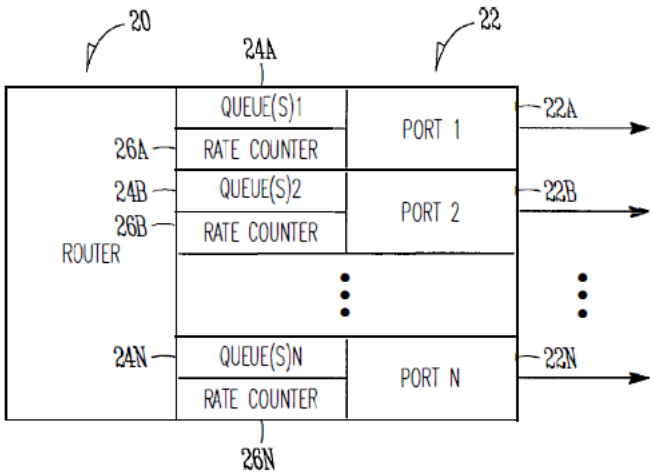
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.") |

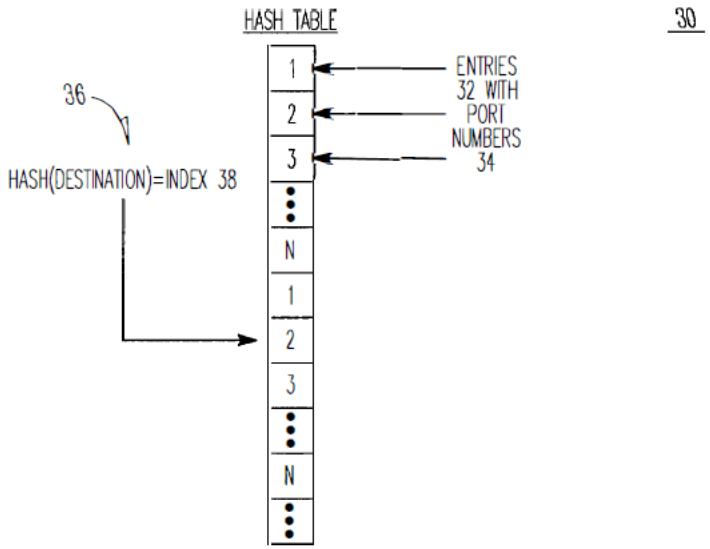
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to |

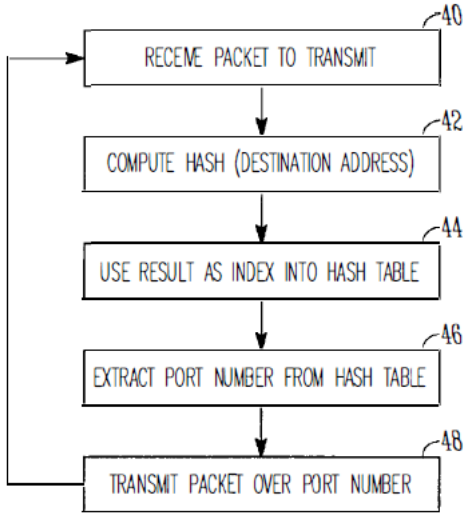
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub- |

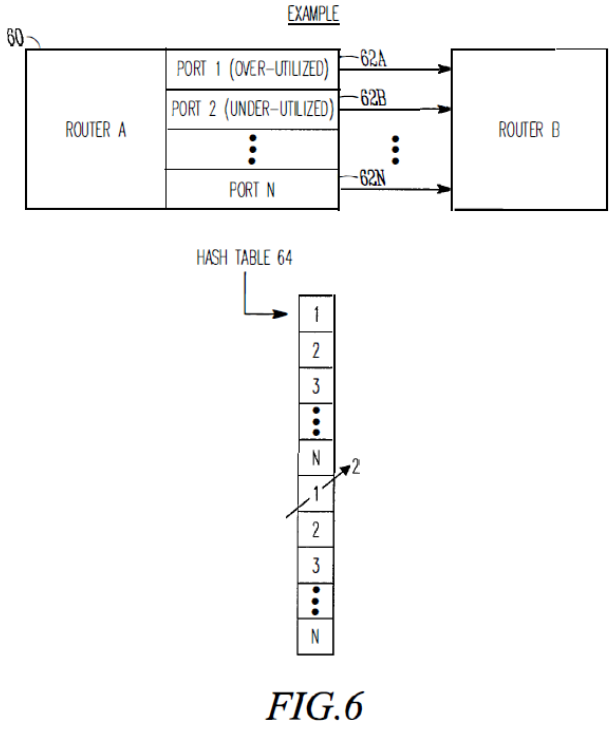
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

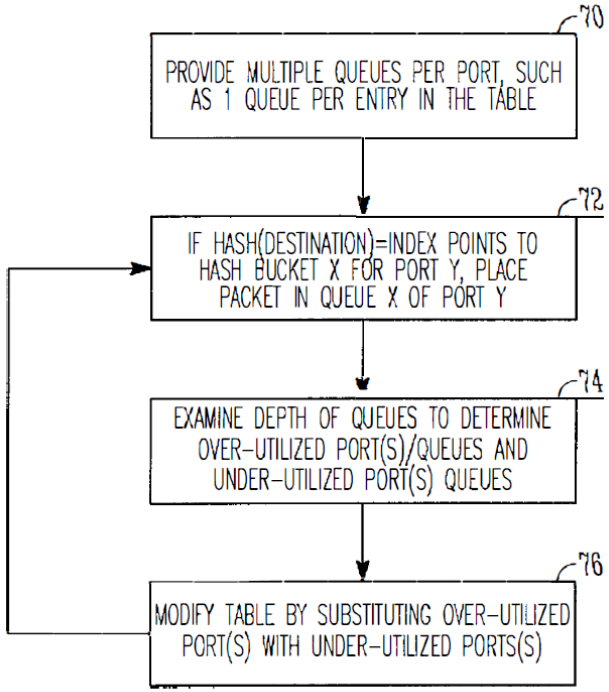
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

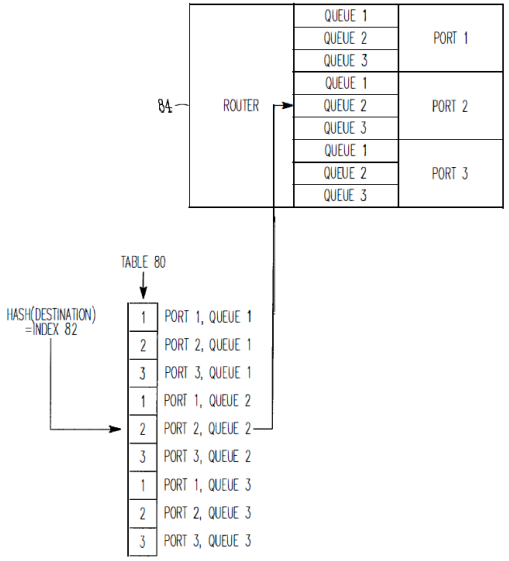
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

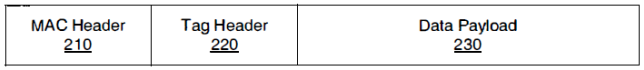
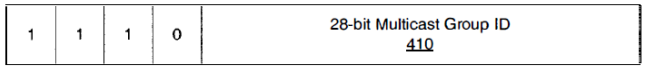
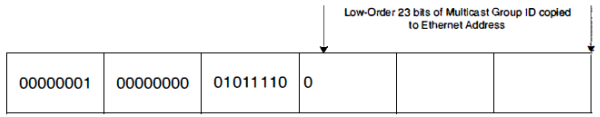
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

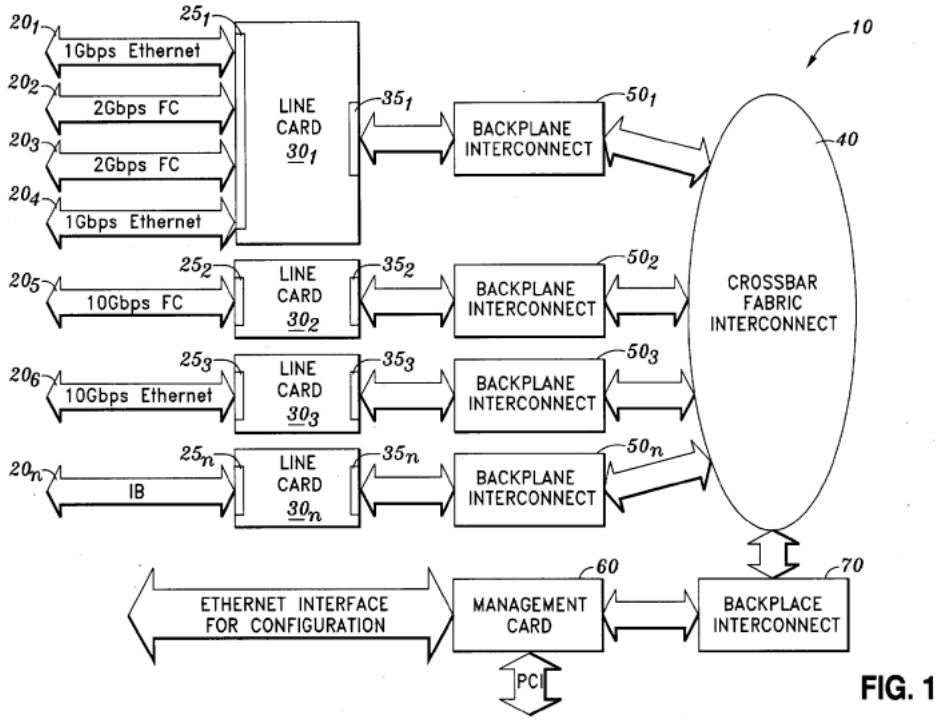
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

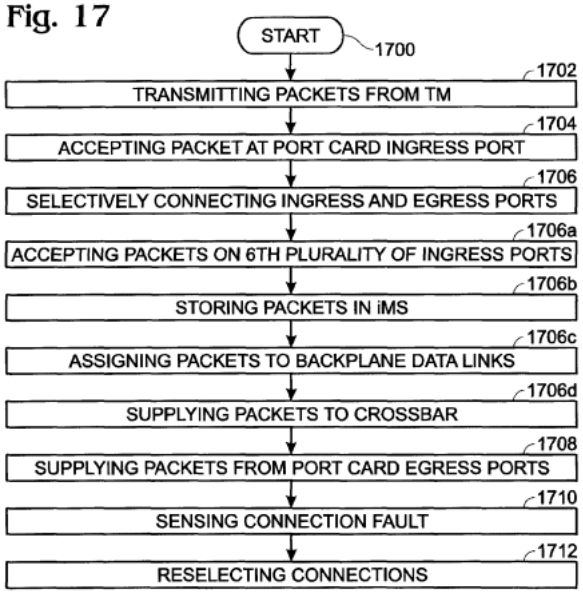
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |
| 15[e] | sending the data frame over the | The Reference discloses sending the data frame over the selected backplane trace. |

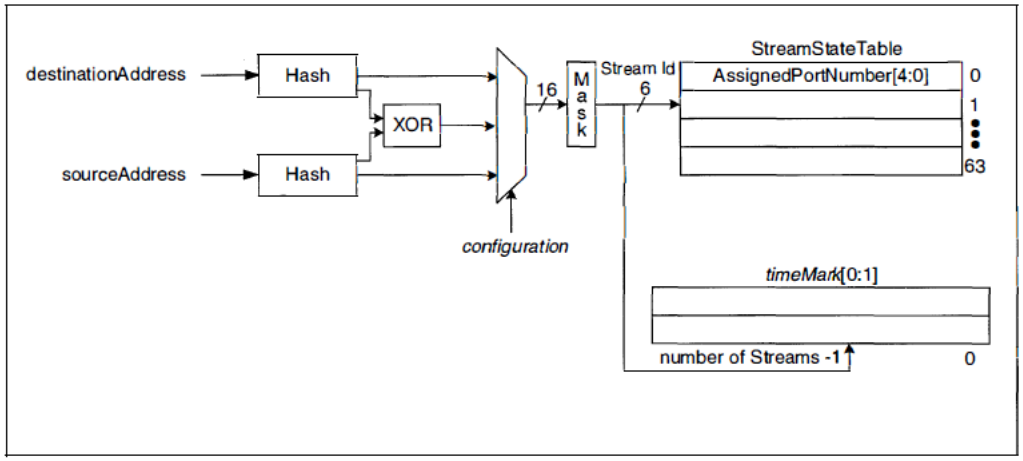
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | selected backplane trace, | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.") |

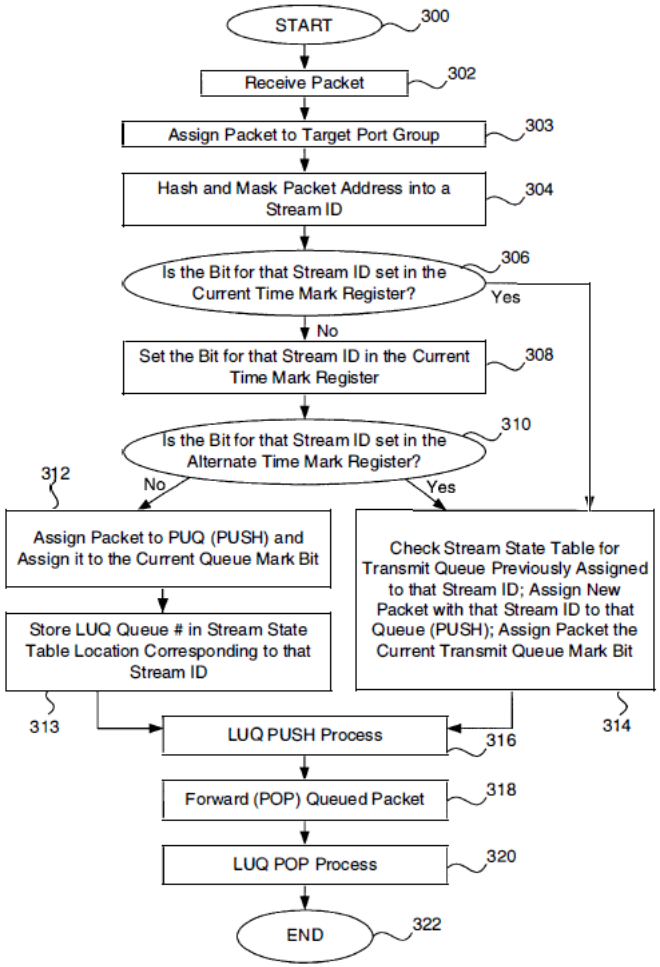
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Viswanathan at Figure 1<br>  <br><br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | substeps. Step 1706e includes each port card accepting packets on a second plurality of port card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | • Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br><br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

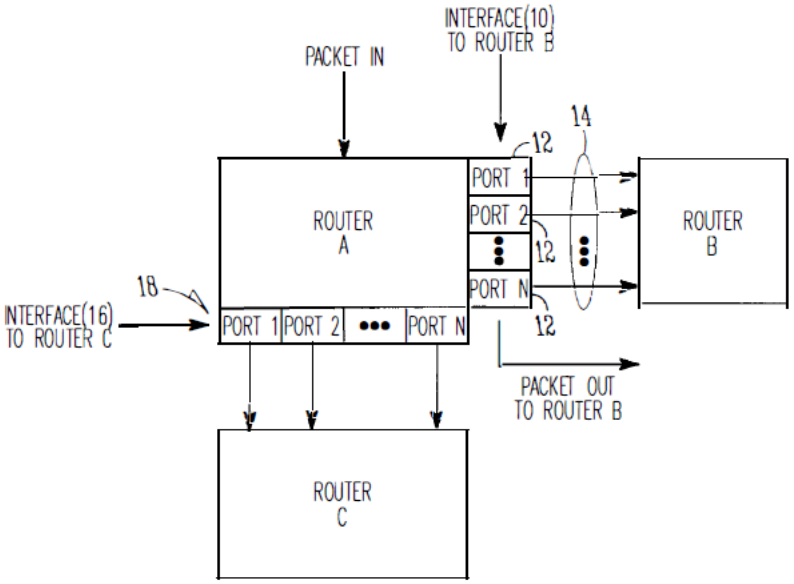
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

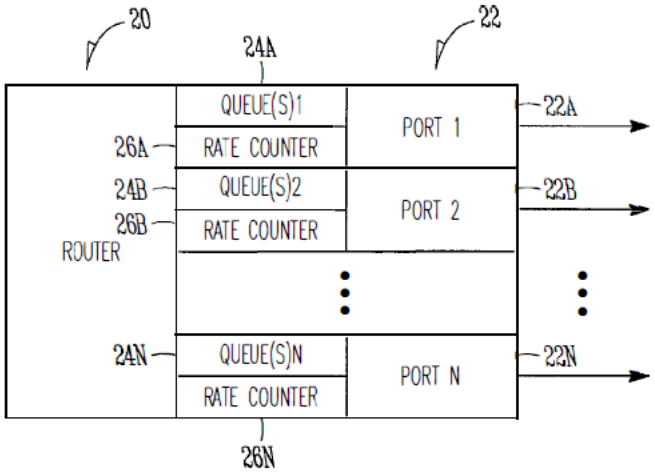
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

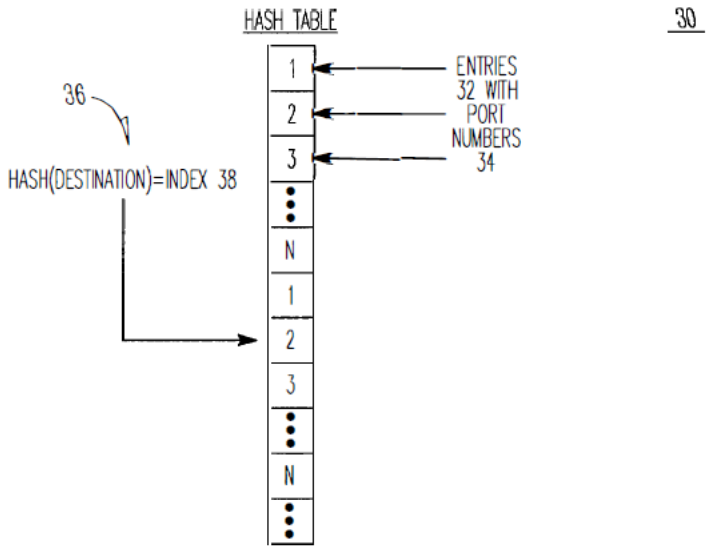
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

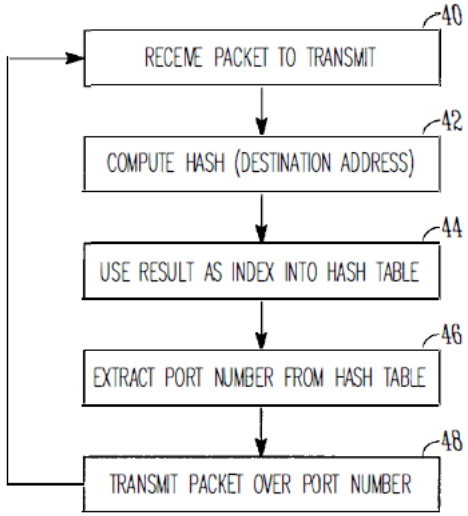
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

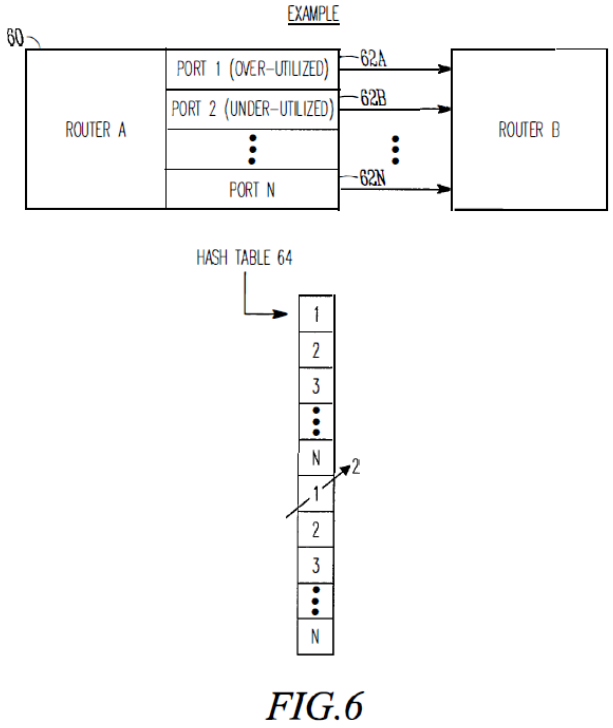
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

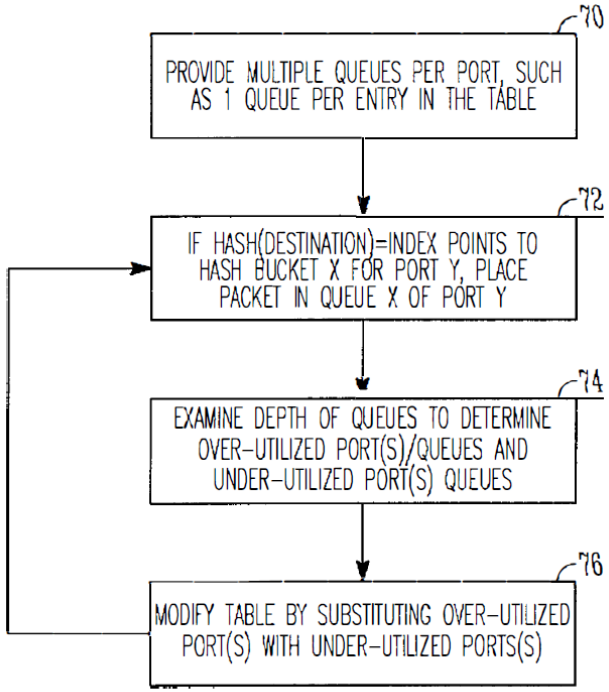
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

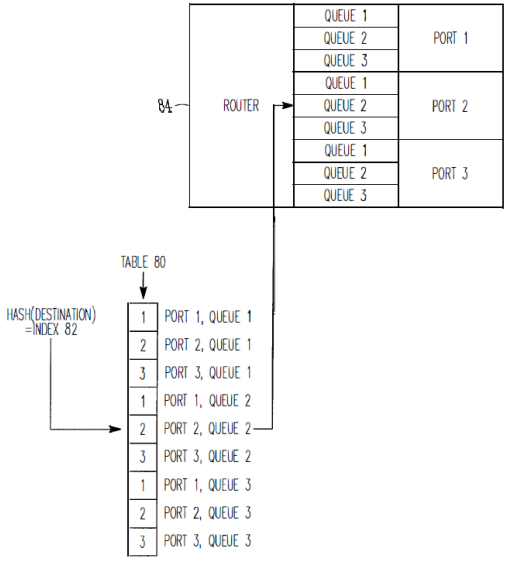
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

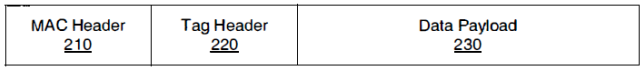
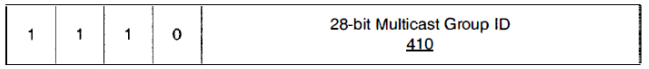
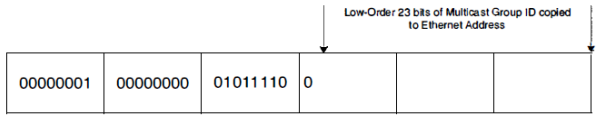
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

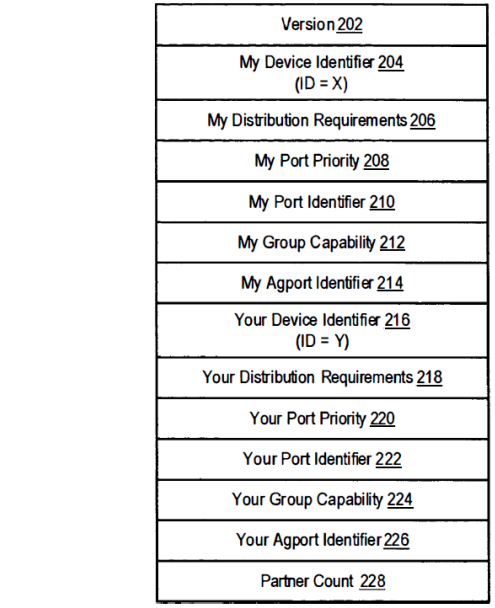
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| 15[f] | at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. | The Reference discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below is an example.<br><br>Smith '430 at 5:51-64 ("The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be oper-ated as a single logical link, referred to herein as a virtual link bundle. Network device 120(1) operates the two links cou-pling network device 120(1) to virtual network device 202 as 55 a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corre-sponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 60 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, virtual link bundles 250(1) and 250(2) are each operated as an EtherChannel™ or as an aggregated link (as described in IEEE 802.3).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the link aggregation technique. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914 |

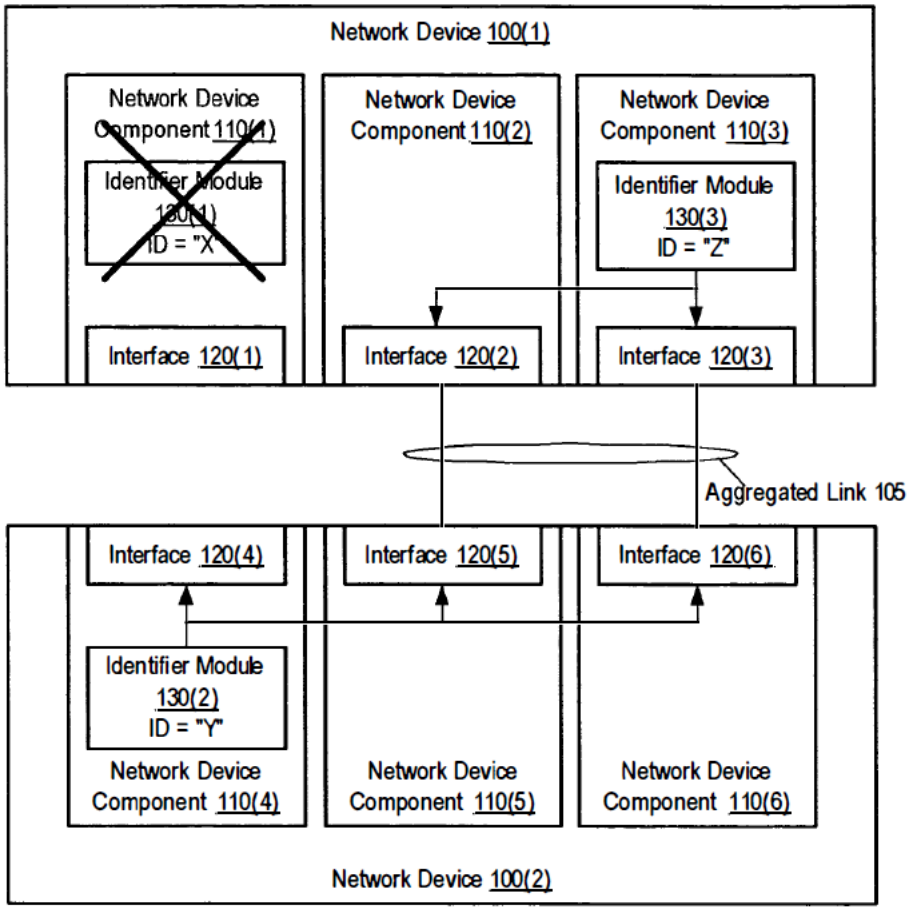
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | • Borgione '125<br><br>DeJager '424 at Abstract ("Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.")<br><br>DeJager '424 at 1:38-50 ("One way to relieve this bottle-neck is to provide a logical grouping of multiple ports into a single port. The bandwidth of the new port is increased since it has multiple lines (cables) connecting a switch and another network device, each line capable of carrying data at the same rate as the line connecting data sources to the switch. This grouping of ports is sometimes referred to as a port aggregation or port group. One example of such a port aggregation implementation is Cisco Technology, Inc.'s Fast EtherChannel™ port group in a Fast Ethernet network. Further information regarding Fast EtherChannel™ may be found on Cisco Technology, Inc.'s World Wide Web site www.cisco.com. This information is incorporated by reference herein for all purposes.")<br><br>DeJager '424 at 2:47-65 ("The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current |

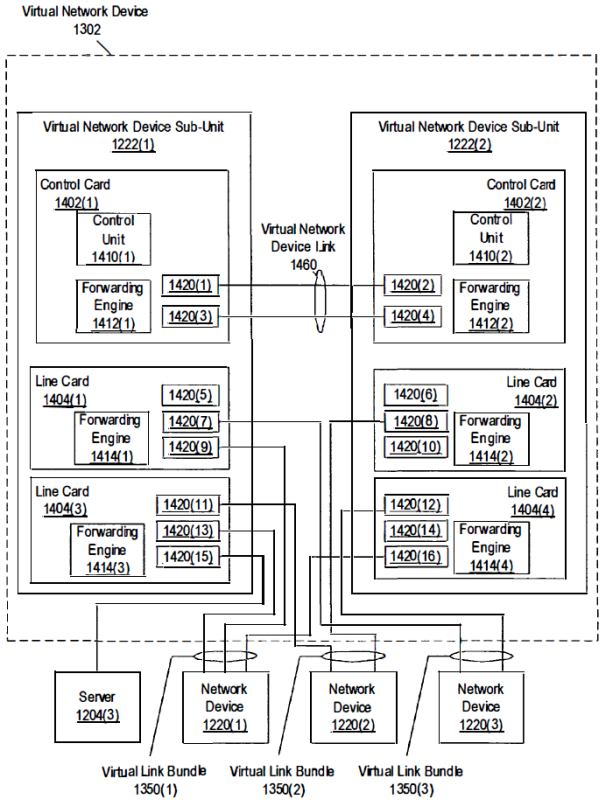
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.")<br><br>DeJager '424 at 2:67-3:15 ("In one aspect, the present invention provides a method of distributing traffic over a network port group. The method involves receiving a packet of data to be forwarded, deter-mining a stream ID for the packet, and determining whether a prior packet having that stream ID has been distributed to a queue on a port in the group during a predetermined time interval. Where a prior packet having that stream ID has not been distributed to a queue on a port of the group during the predetermined time interval, the method involves allocating the packet to a queue of a port having a lesser load in its queue than a queue of any other port of the group. The method may also involve, where a prior packet having that stream ID has been distributed to a queue on a port of the group during the predetermined time interval, allocating the packet to that queue. In addition, the method may involve monitoring the port group queues to maintain proper iden-tification of the least utilized queue.")<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 4:29-46 ("The present invention provides methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present 432ubsti-tion is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other network environments.")<br><br>DeJager '424 at 4:47-58 ("FIG. 1 illustrates a block diagram of a simple network. The network 100 includes two servers S1. And S2, respectively, and two switches, X1 and X2, respectively, as well as four clients C1, C2, C3 and C4, respectively. Clients C , , and C4 are connected to switch X1 by, for example, Fast Ethernet links 102 via ports 1, 2, 3 and 4, respectively. Server S1 is connected to switch X1 via a port aggregation 104, which is a port group composed of ports 5 and 6 of switch X1 . Switch X1 is connected to switch X2 via a second port aggregation 106 which includes ports 7, 8 and 9. Switch X2 is connected to server S2 via port O and Fast Ethernet link 108.")<br><br>Dontu at Abstract ("Various methods and systems for preventing erroneous link aggregation due to component relocation are disclosed. Such methods include a method for changing the identifier used by a network device and communicating the identifier change to a peer |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | network device without disrupting an aggregated link. In one embodiment, a method involves detecting an identifier change and sending a Port Aggrega-tion Protocol (PagP) protocol data unit (PDU) that includes a new identifier and information. The information indicates the identifier change. The new identifier identifies a network device subsequent to the identifier change. Another embodi-ment of a method involves detecting an identifier change and, subsequent to the identifier change, sending a link aggregation protocol PDU that includes an "old device identifier" field dedicated to conveying an old identifier. The old identifier identifies a network device prior to the iden-tifier change.")<br><br>Dontu at Figure 2 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Dontu at Figure 3 |

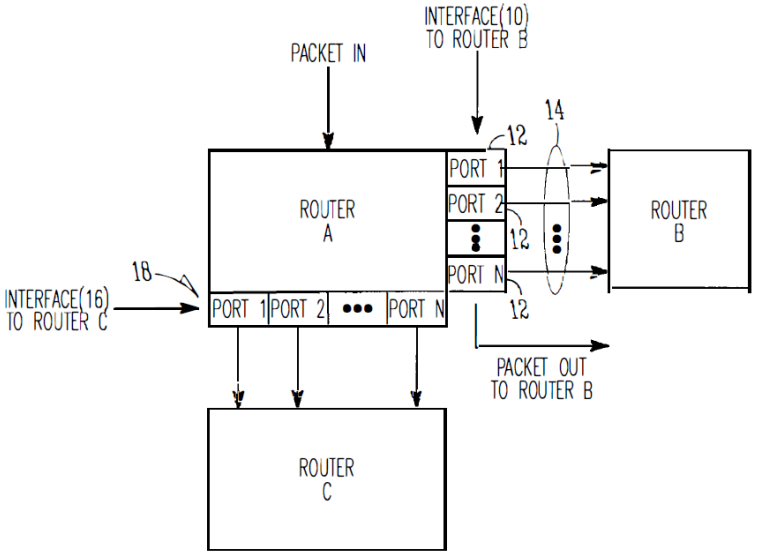
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Dontu at Figure 14 |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | |  <br>FIG. 14<br><br>Dontu at [0004] ("Link aggregation is used to logically combine two or more individual links into a single aggregated link. Link aggregation can provide improved performance and increased fault tolerance. Improved performance arises because the aggregated link appears to have a bandwidth equal to the combined bandwidth of the individual links. Traffic can be load-balanced among the individual links. Increased fault tolerance is provided since one or more individual links within an aggregated link can fail without disrupting communication between the devices coupled by the aggregated link. Link aggregation techniques include |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Link Aggregation Control Protocol (LACP), which is defined in IEEE 803.2ad, and Port Aggregation Protocol (PagP), which is a standard promulgated by CISCO SYS-TEMS, INC.")<br><br>Dontu at [0012] ("The method can also involve detecting whether a partner interface is executing a compatible version of PagP. If the partner interface is not executing the compatible version of PagP, the compatible version of PagP can be provided to the partner interface. Alternatively, if the partner interface is not executing the compatible version of PagP, the partner interface can be inhibited from including a link in an aggregated link.")<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).") |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Dontu at [0037] ("In this example, the network devices 100(1) and 100(2) use Port Aggregation Protocol (PagP) to form aggre-gated links. Network devices 100(1) each send PagP pro-tocol data units (PDUs) to each other in order to determine whether any of the links between the two network devices can be combined into an aggregated link. Each PagP PDU includes an identifier that uniquely identifies the network device that sent that PagP PDU. Within network device 100(1), identifier module 130(1) of network device compo-nent 110(1) supplies an identifier "X" to each of the inter-faces 120(1)-120(3) within network device 100(1). Inter-faces 120(1)-120(3) include identifier X in each PagP PDU sent by those interfaces. Similarly, identifier module 130(2) of network device component 110( 4) supplies an identifier "Y" to each interface 120( 4)-120( 6) of network device 100(2). Interfaces 120( 4)-120( 6) include identifier Yin each PagP PDU sent by those interfaces.")<br><br>Dontu at [0040] ("FIG. 2 illustrates some of the fields that can be included in a PagP PDU. As shown, PDU 200 includes Version field 202, My Device Identifier field 204 ("My" refers to the device sending the PagP PDU), My Distribu-tion Requirements field 206, My Port Priority field 208, My Port Identifier field 212, My Group Capability field 212, My Agport (Aggregated Port) Identifier field 214, Your Device Identifier field 216 ("Your" refers to the device to which the PagP PDU is being sent), Your Distribution Requirements field 218, Your Port Priority field 220, Your Port Identifier field 222, Your Group C"pabi"ity field 224, Your Agport Identifier field 226, and Partner Count field 228.")<br><br>Dontu at [0110] ("Interfaces 1420(13), 1420(9), and 1420(16), which are each coupled to network device 1220(1) by virtual link bundle 1350(1), form an interface bundle (e.g., an Ether-Channel I port bundle). Similarly, interfaces 1420(11) and 1420(8) form another interface bundle that is coupled to network device 1220(2) by virtual link bundle 1350(2). Interfaces 1420(7) and 1420(12) form a third interface bundle that is coupled to network device 1220(3) by virtual link bundle 1350(3). Within virtual network device 1302, each interface in the same interface bundle is assigned the same logical identifier. For example, interfaces 1420(13), 1420(9), and 1420(16) are each assigned the same logical identifier. In some embodiments, packets received via one of these interfaces are tagged or otherwise associated with the logical identifier to indicate that those packets were received via the |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | virtual link bundle coupling virtual network device 1302 to network device 1220(1). It is noted that similar interface bundles are implemented within each network device 1220(1)-1220(3), and that interfaces included in such bundles are also assigned the same logical identifier by each network device ( or by virtual network device 1302, in embodiments in which virtual network device 1302 controls the configuration of the network devices 1220(1)-1220(3)). For example, network device 1220(1) can assign the same logical identifier to each of the interfaces coupled to virtual link bundle 1350(1).")<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 2:6-22 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.") |

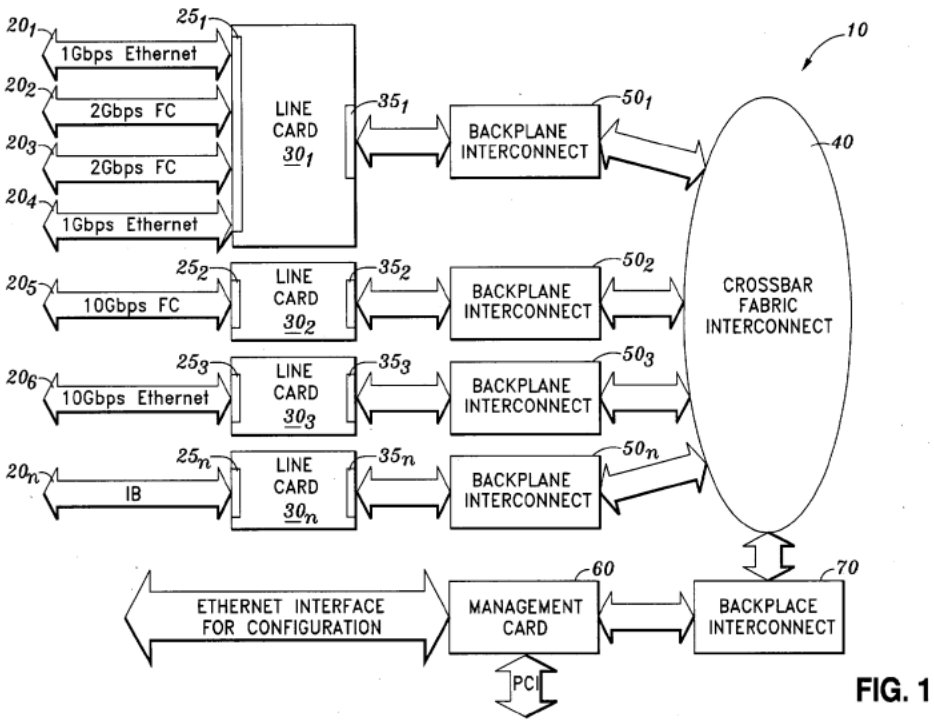
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Li '914 at 4:9-25 ("Referring to FIG. 1, a Router A is shown having an inter-face 10 with a plurality of ports which connect Router A with Router B over a plurality of connections, lines, wires, links or bundled links 14. The ports 12 of Router A are configured to permit transmission of packets from Router A to Router B, and these ports 12 may be referred to as output ports, egress ports, links, or the like. As shown in FIG. 1, port 1 to port N may be connected with Router B, and there may be additional interfaces 16 having ports 18 connected with other routers in the network. When a packet is received by Router A, Router A determines whether the received packet should be transmitted to Router B or to other routers connected to Router A, based in part upon the destination address of the packet. If a packet is to be transmitted from Router A to Router B, then Router A may transmit this packet over the one of the ports 12 shown in FIG. 1.")<br><br>Li '914 at Figure 1<br>  <br><br>*FIG. 1* |

| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | Borgione '125 at 1:55-65 ("Link nodes 110 and 120 can be in physically remote locations, thereby connecting their associated local area networks (LANs). The plurality of network links 150 between link nodes 110 and 120 can be aggregated as a single logical link over which all traffic between link nodes 110 and 120 is distributed. Such aggregation multiplies the available bandwidth for communications between link nodes 110 and 120, and therefore between the two local area networks. When appropriately configured, such a connection can permit the two local area networks to interact as if they were one large local area network.")<br><br>Borgione '125 at 1:66-2:7 ("As stated above, the plurality of network links between 110 and 120 can be aggregated as a single logical link. In this manner, each link node 110 and 120 sees the plurality of network links between them as one logical interface. One type of such an aggregate of links is an EtherChannel, a protocol that allows up to eight Fast Ethernet or Gigabit Ethernet links to be aggregated. Routing protocols treat the aggregated links as a single, routed interface with a common IP address.")<br><br>Borgione '125 at 5:28-50 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even |

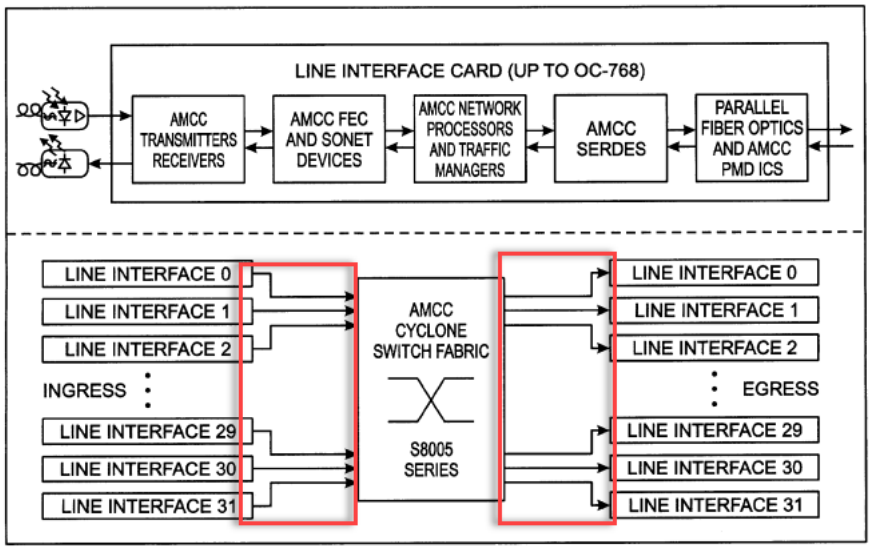
| No. | '740 Patent Claim 15 | The Reference |
|---|---|---|
| | | distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.") |

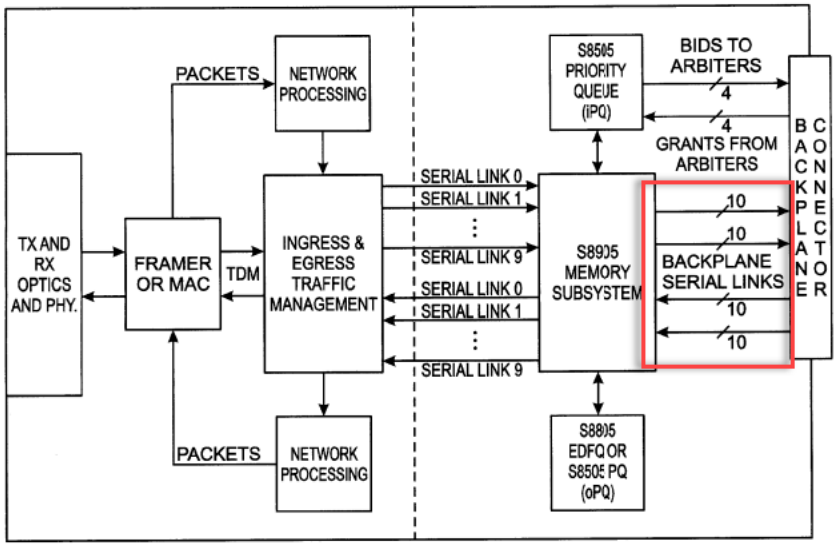
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| 16 | The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. | The Reference discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be implemented as one or more line cards in a networked environment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collectively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar interconnect 40 via backplane interconnects 501-50n (collectively, "backplane interconnects 30"). It should be appreciated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communication across all connected systems using a fabric |

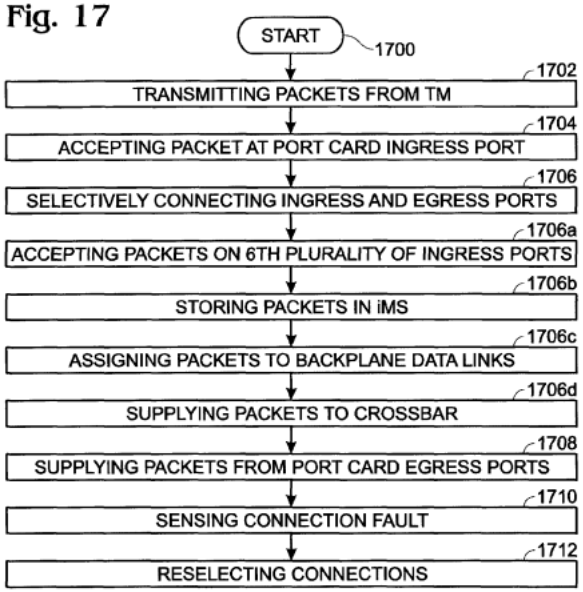
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus architecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress |

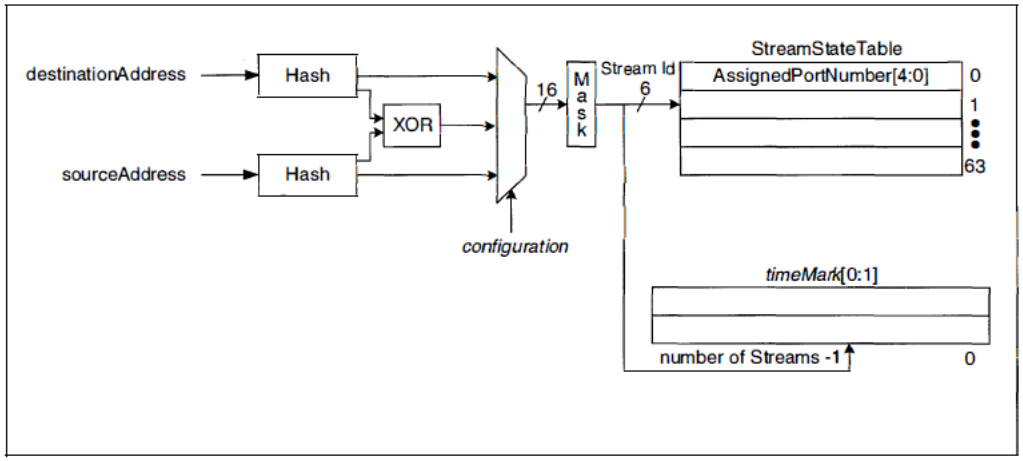
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Functional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

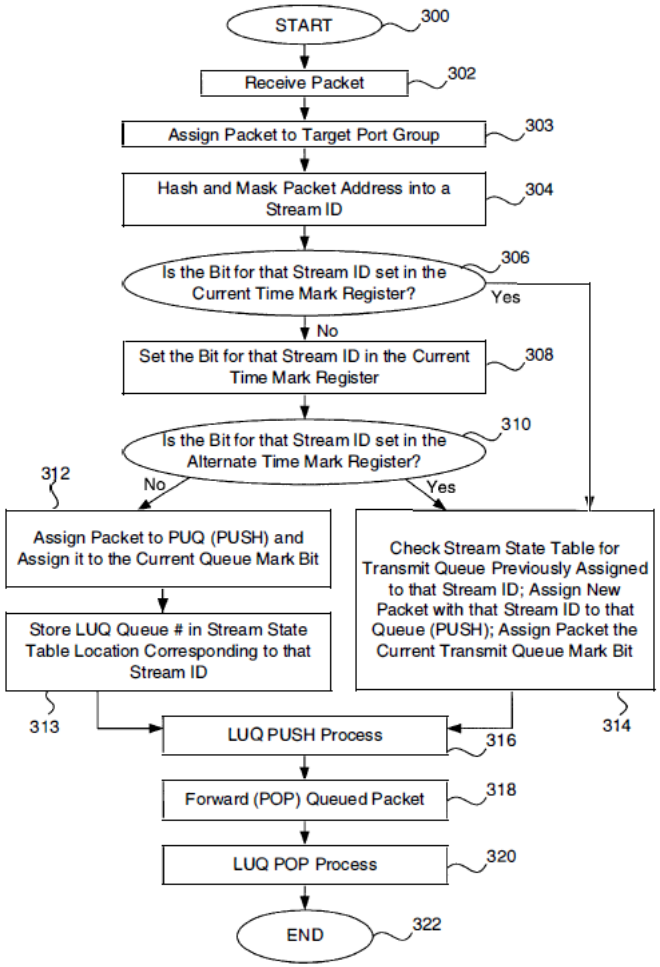
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), network device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

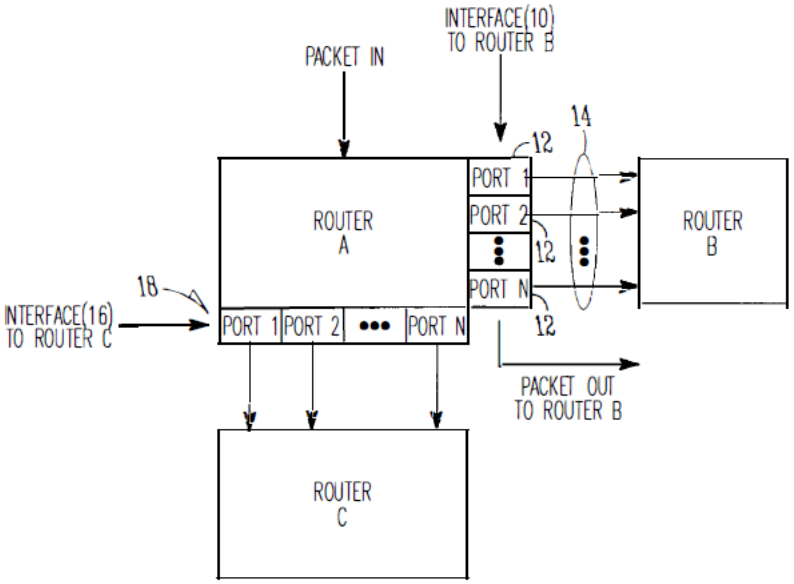
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

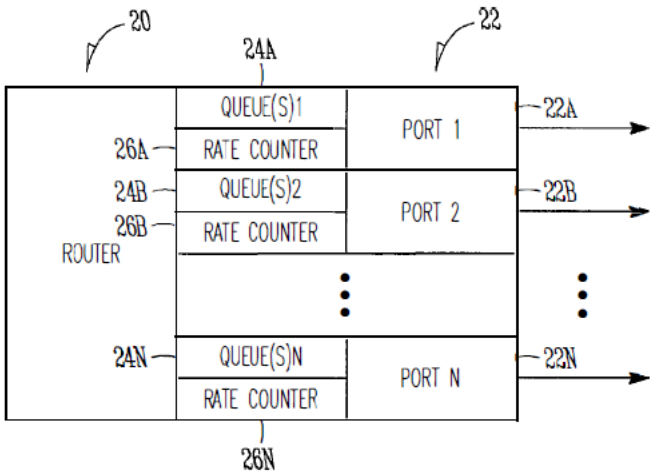
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

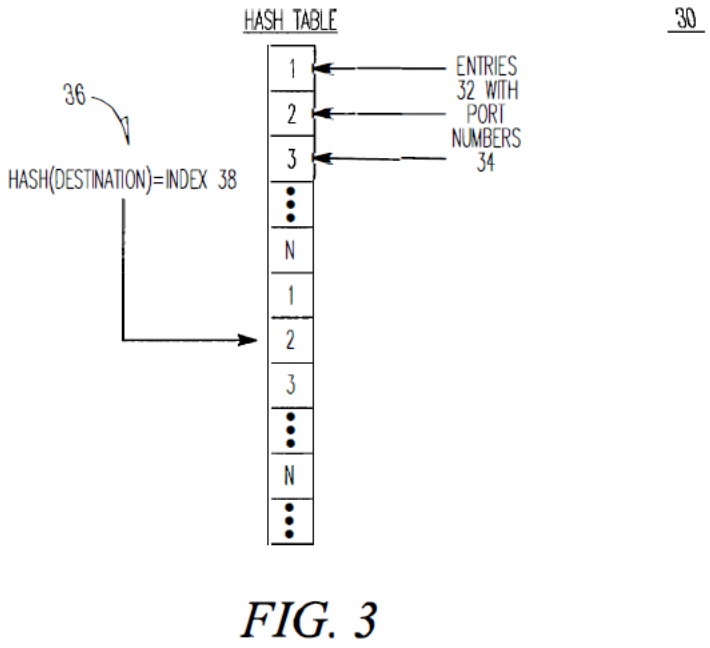
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

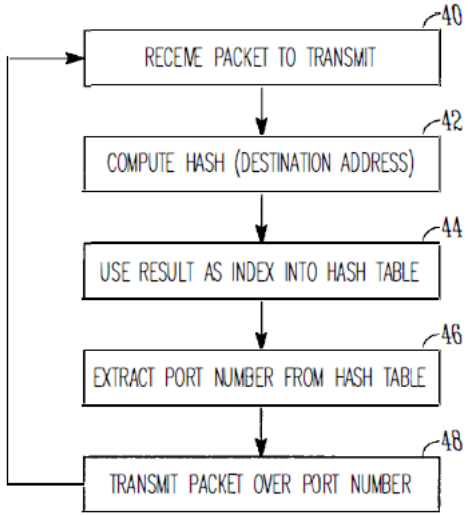
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

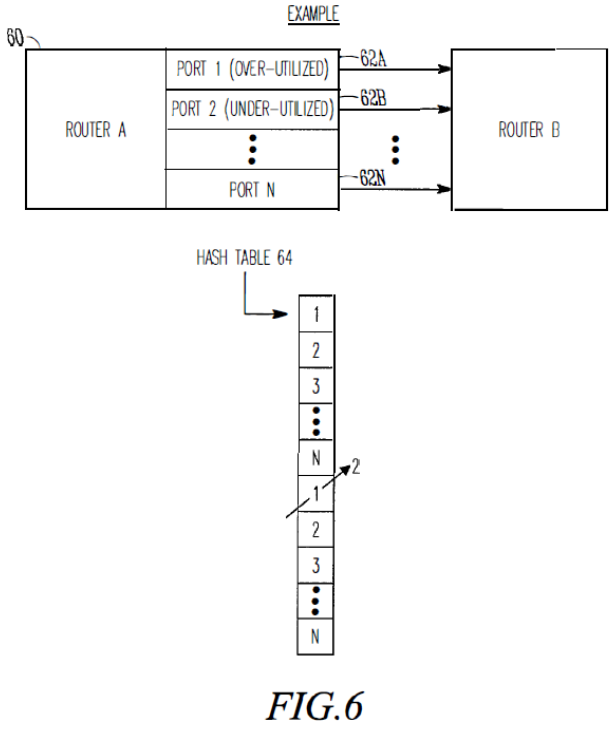
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

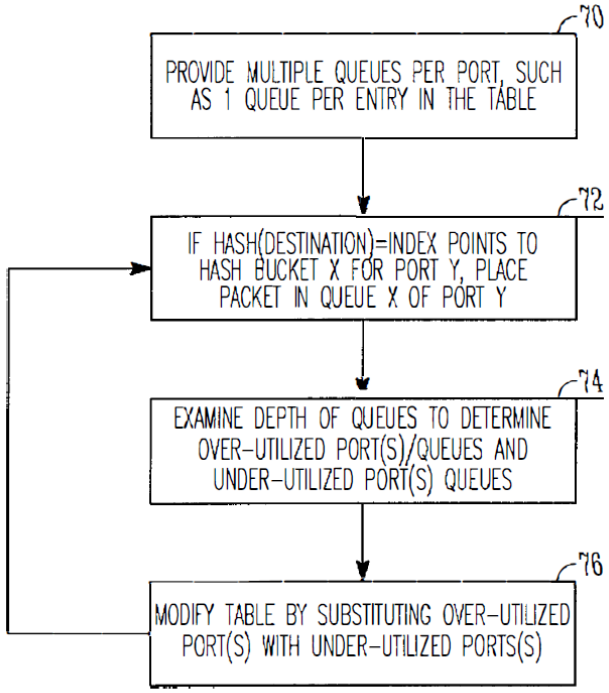
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

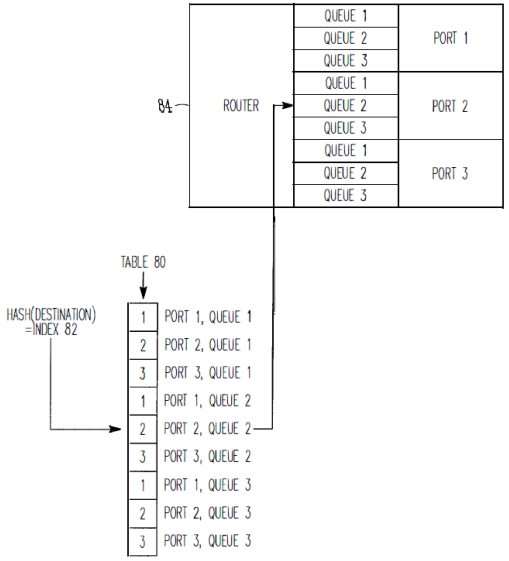
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

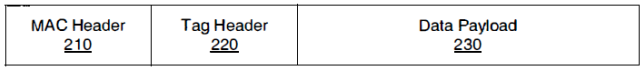
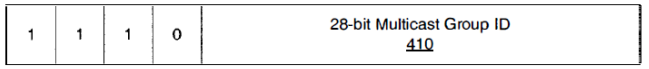
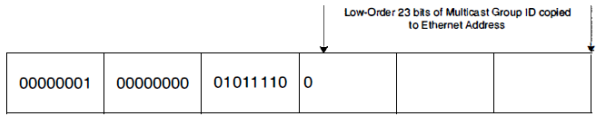
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by 466ubstitute- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

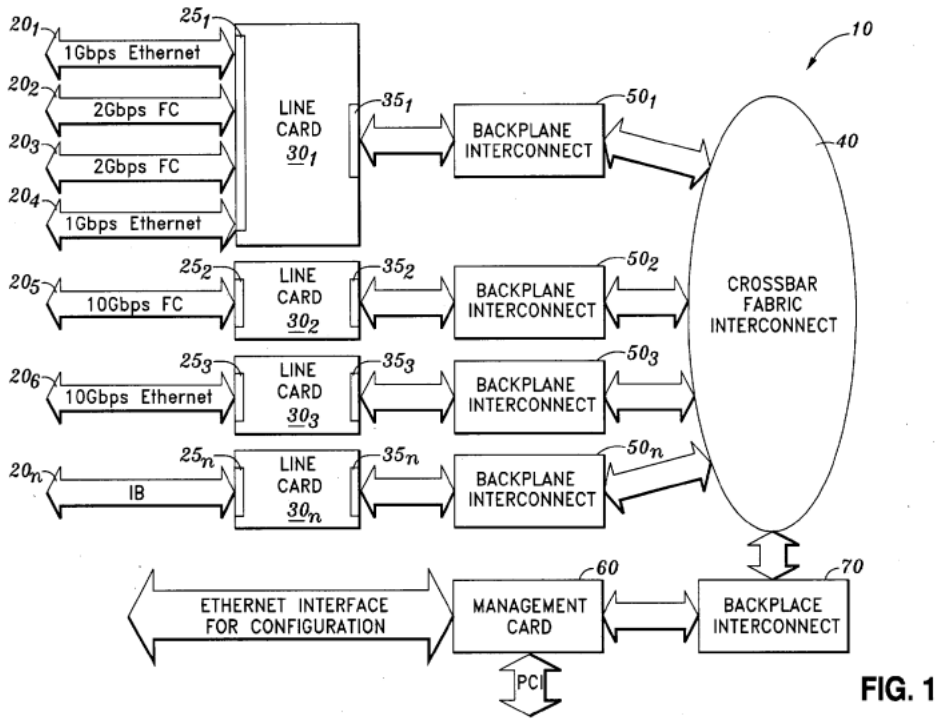
| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 16 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

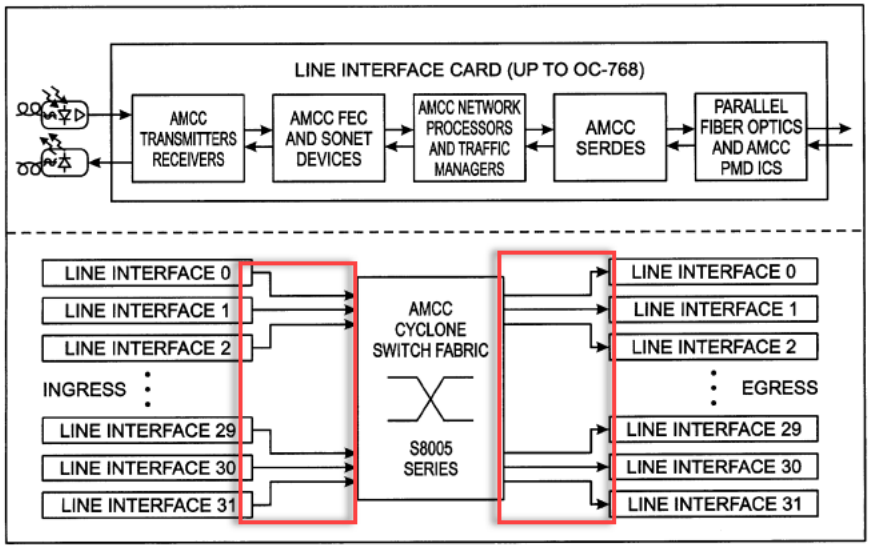
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| 17[preamble] | Apparatus for connecting a network node with a communication network, comprising: | The Reference discloses apparatus for connecting a network node with a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 17[a] | one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network, | The Reference discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 17[b] | at least one of said interface modules being operative to communicate in both an upstream direction | The Reference discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was |

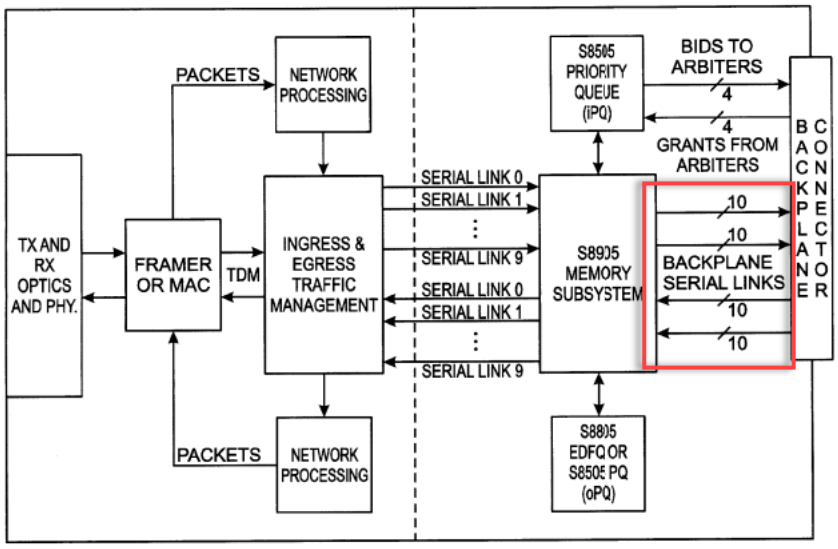
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | and a downstream direction; | known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 17[c] | a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules; | The Reference discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 17[d] | a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and | The Reference discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references. |

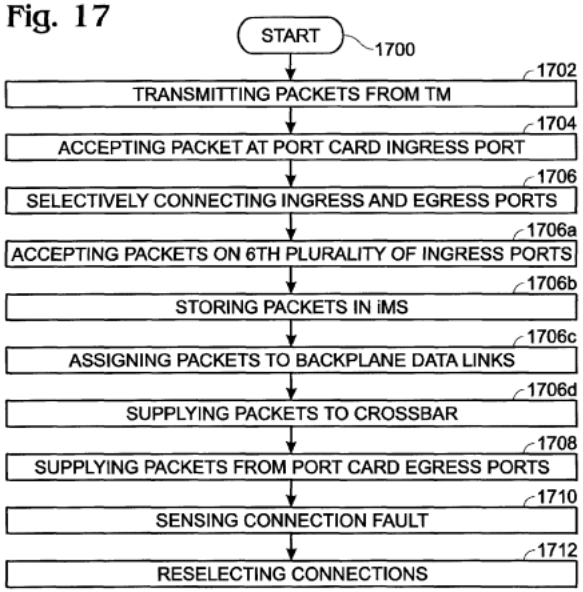
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

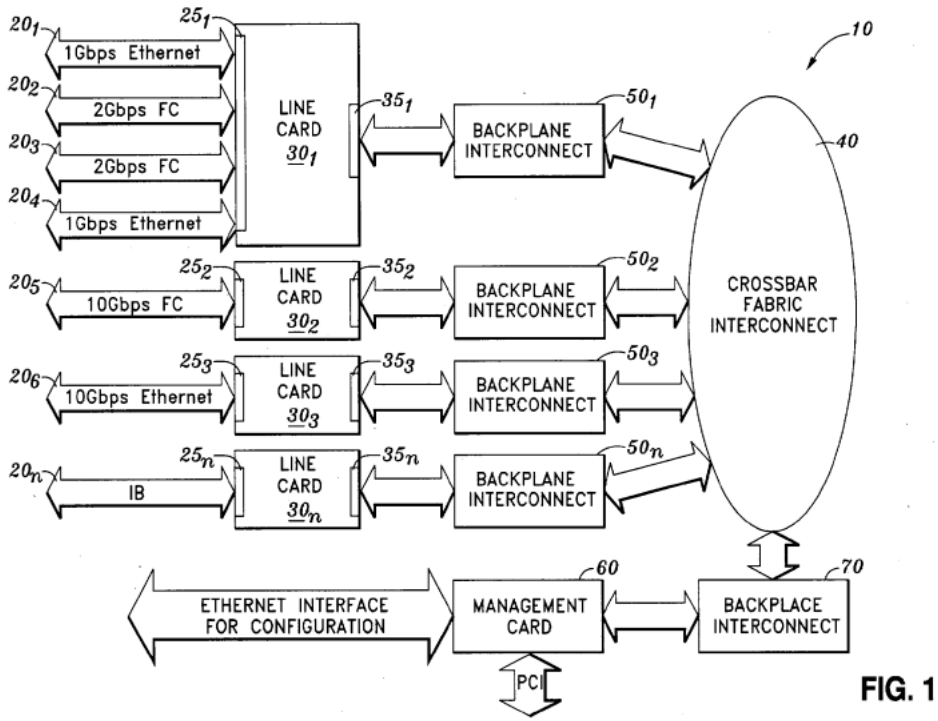
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

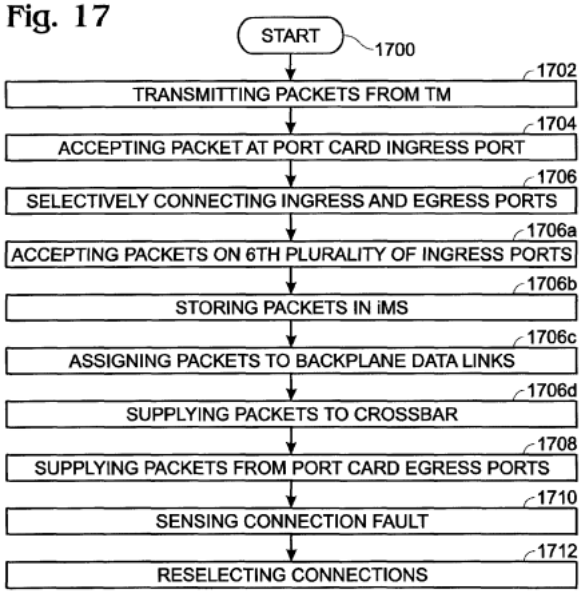
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 17[e] | a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame; | The Reference discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces |

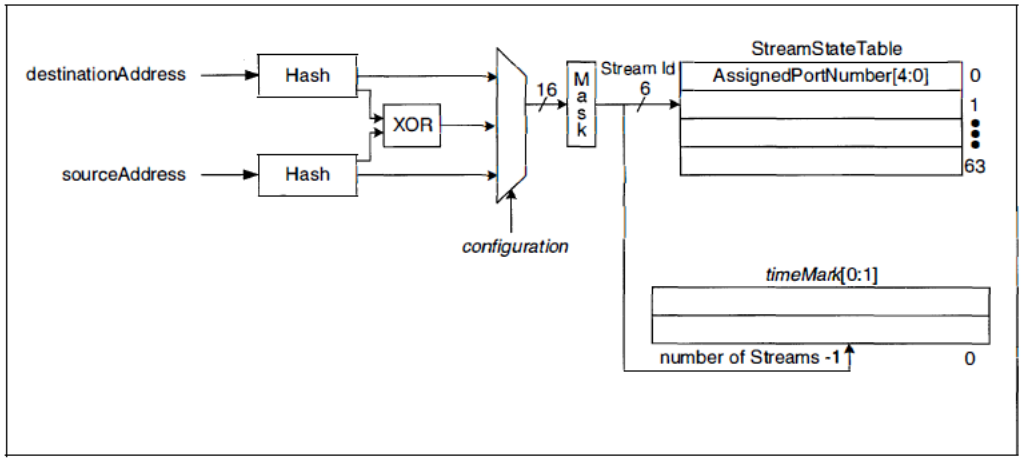
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

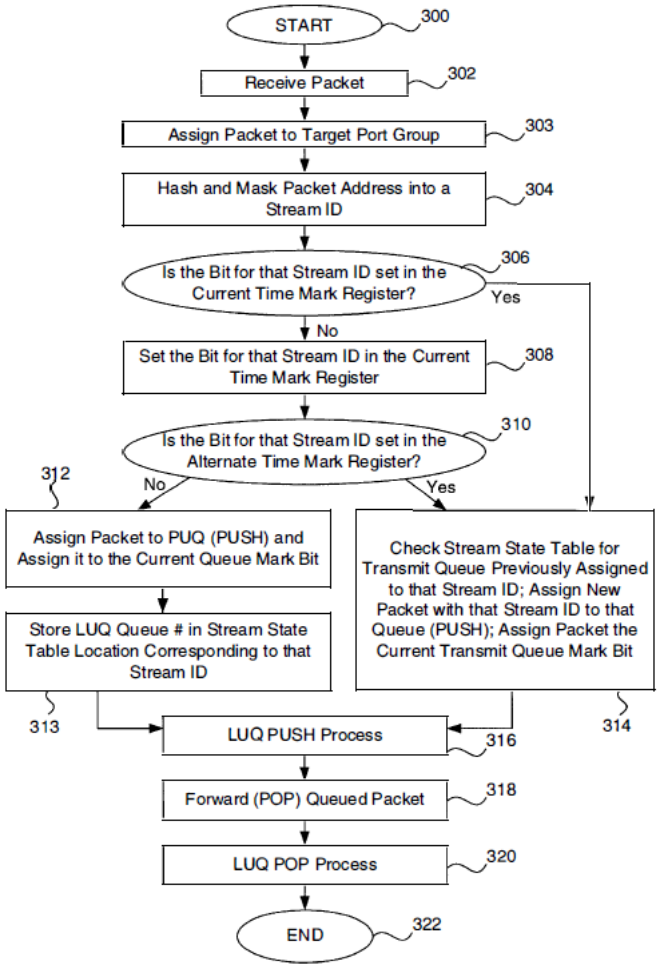
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | • Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

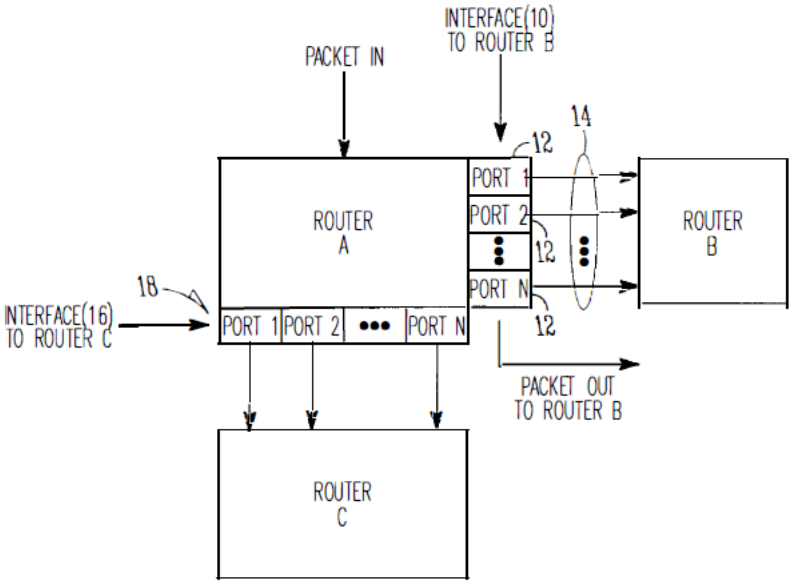
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

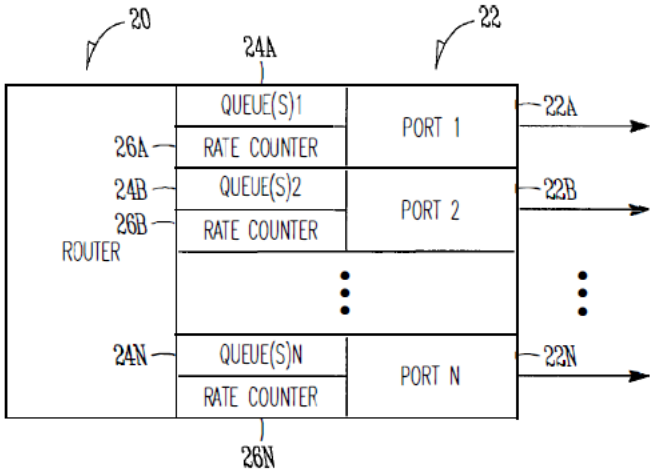
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

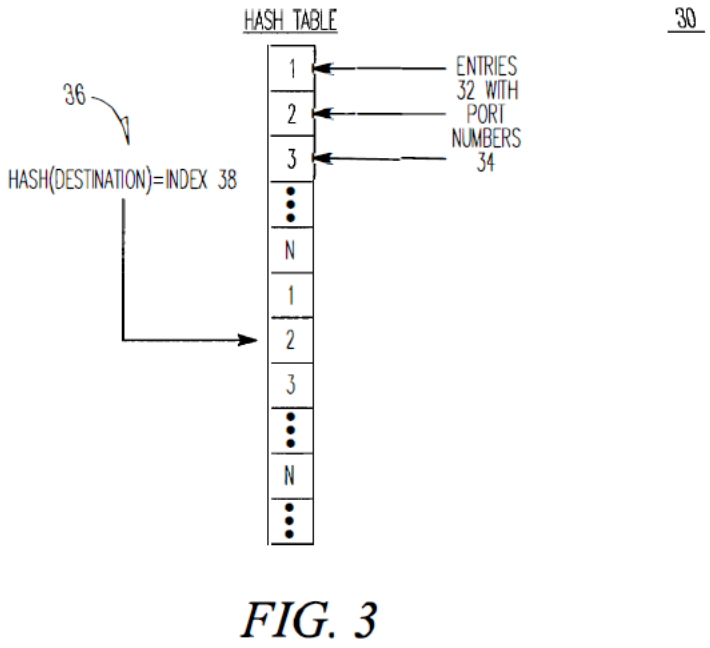
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

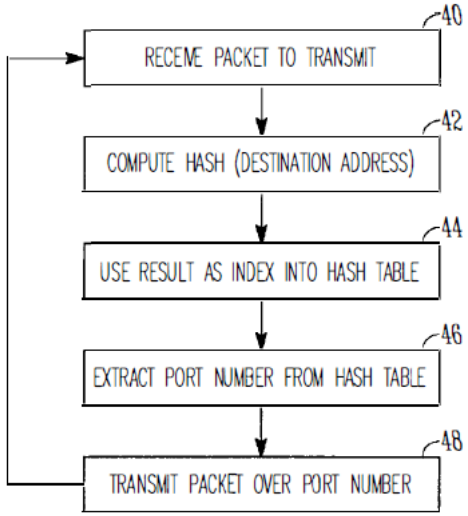
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

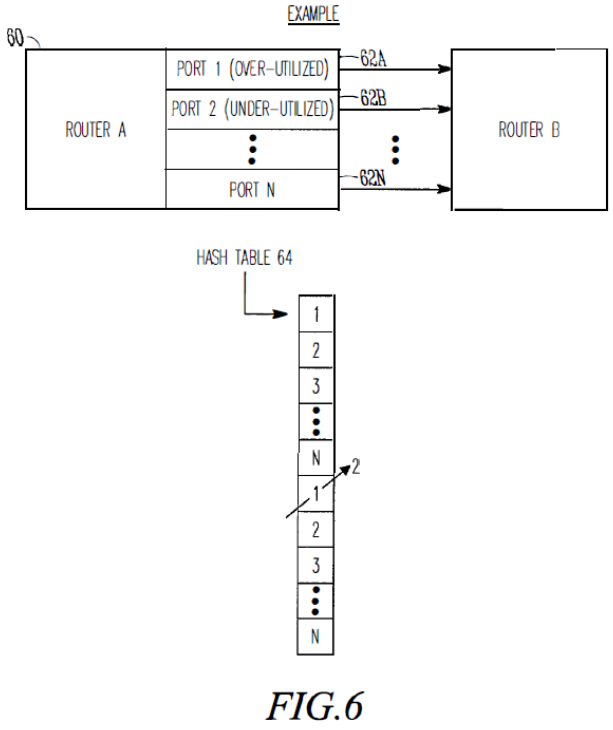
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

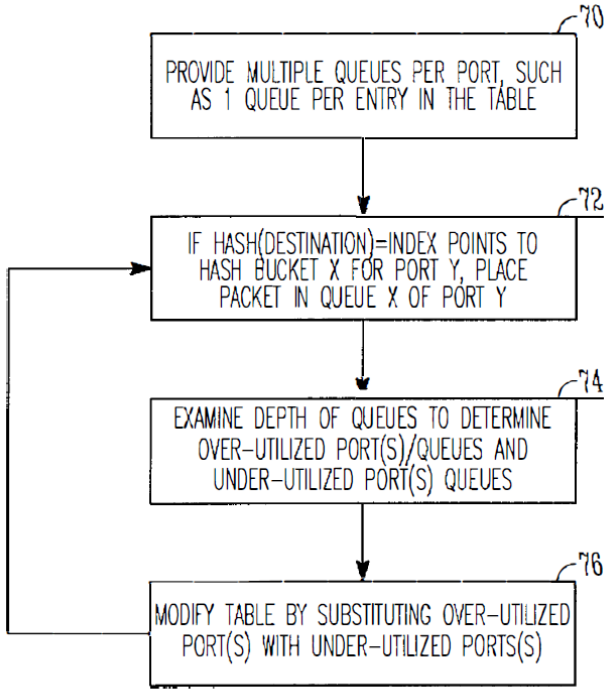
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

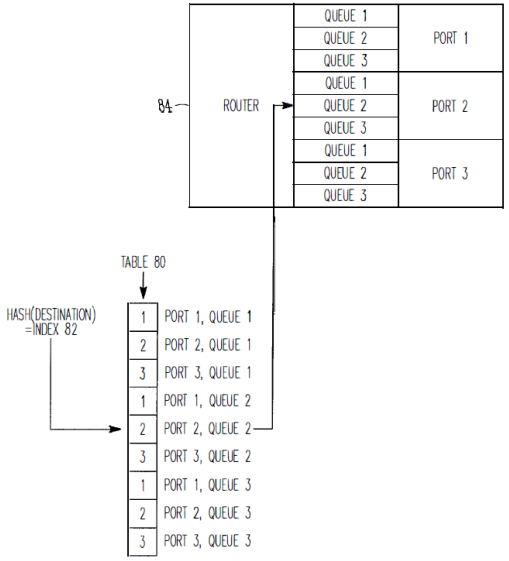
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

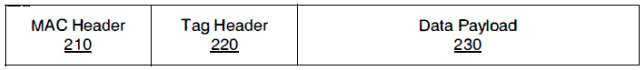
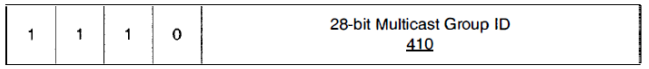
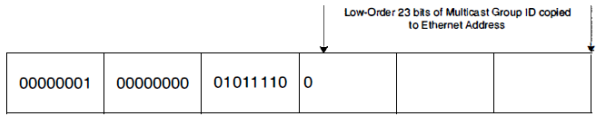
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by 500ubstitute- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

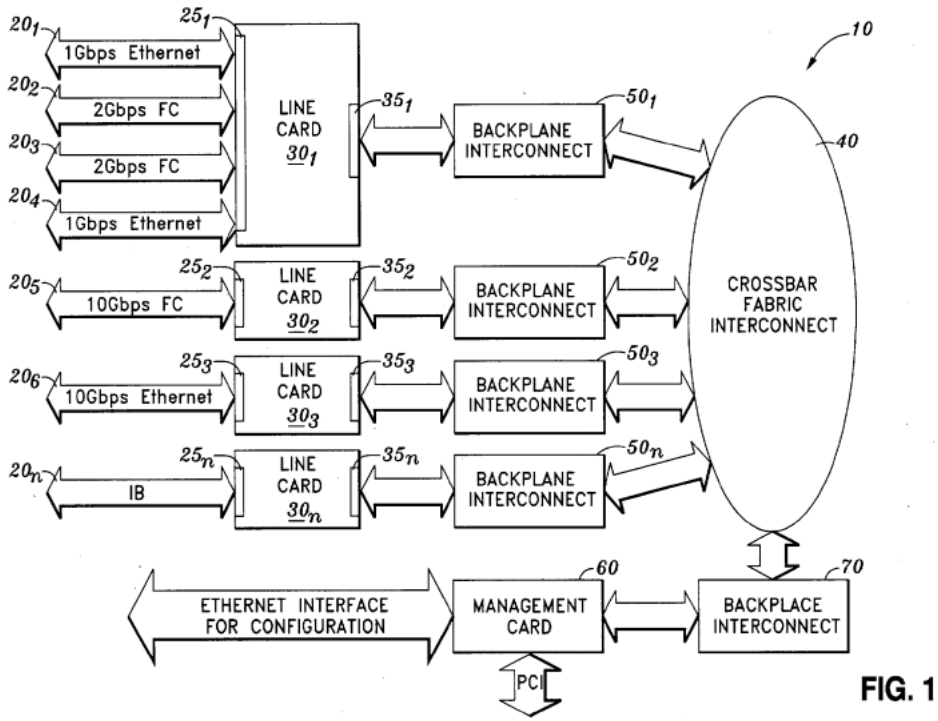
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

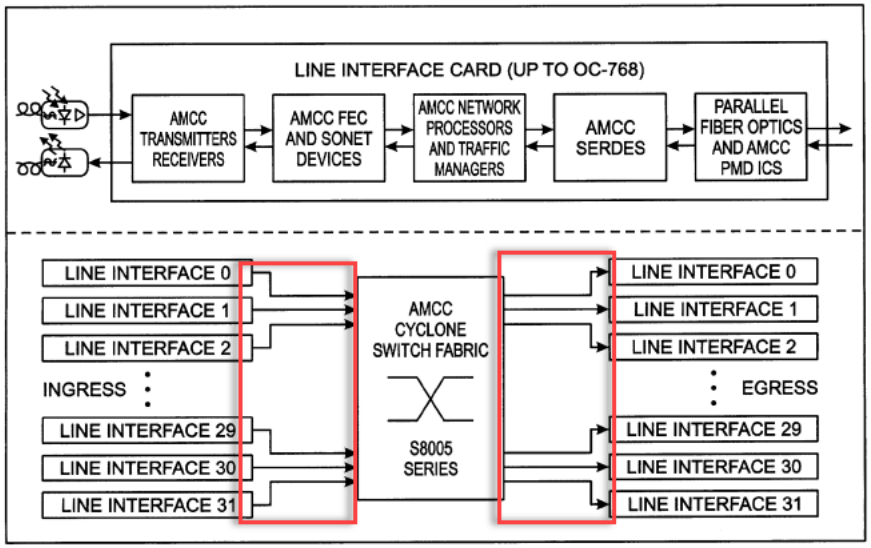
| No. | '740 Patent Claim 17 | The Reference |
|---|---|---|
| 17[f] | at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. | The Reference discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

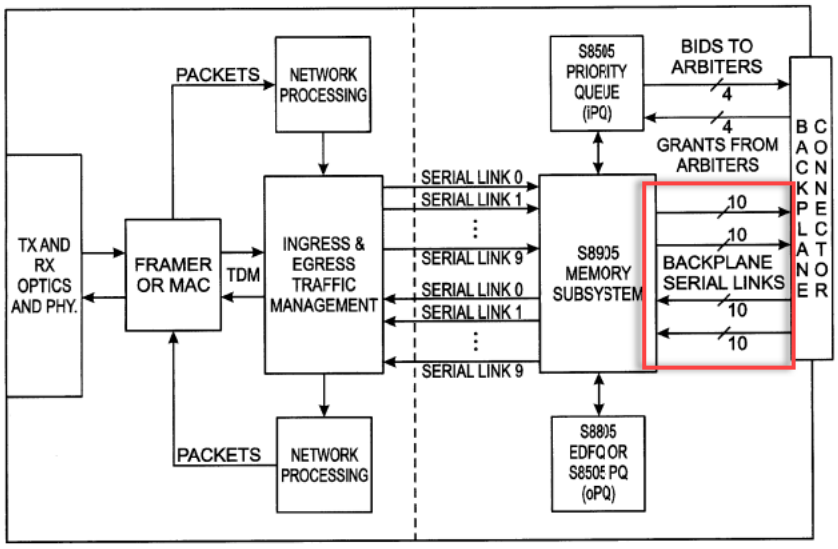
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| 18[a] | The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled, | The Reference discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references. |

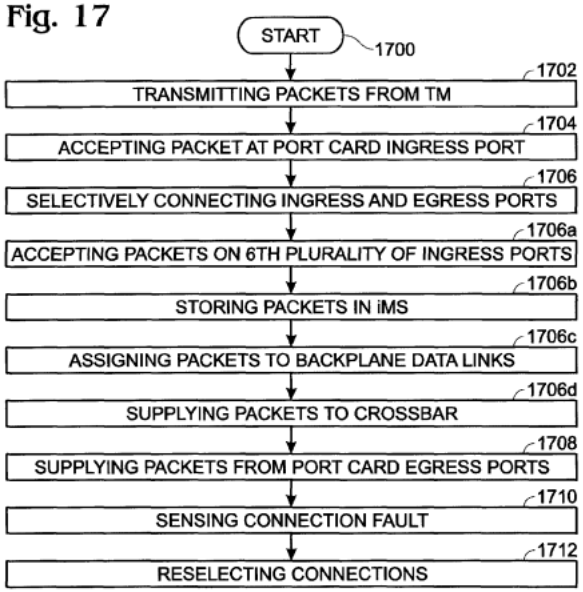
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

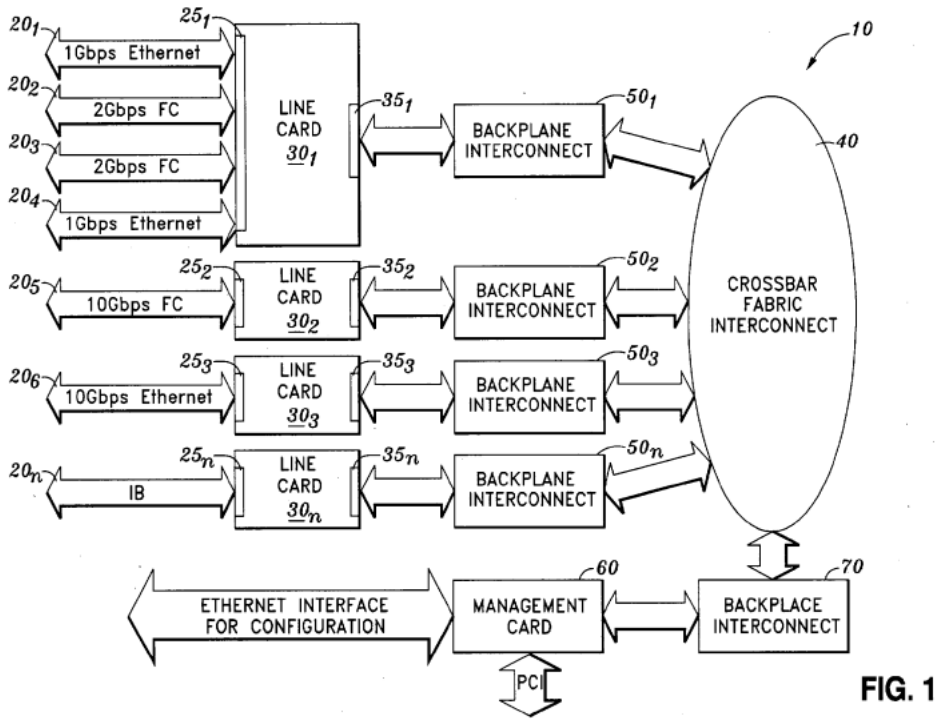
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

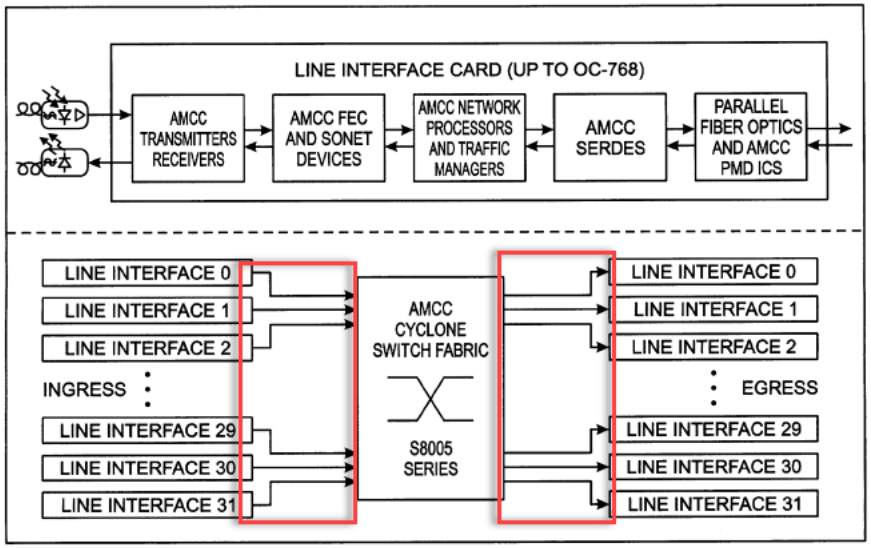
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

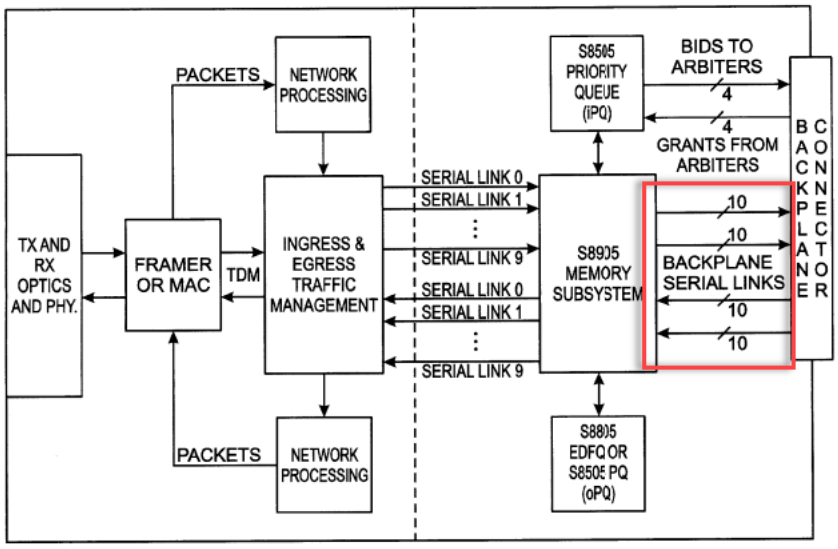
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 18[b] | wherein the second physical links comprise back plane traces formed on the backplane. | The Reference discloses wherein the second physical links comprise back plane traces formed on the backplane.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces |

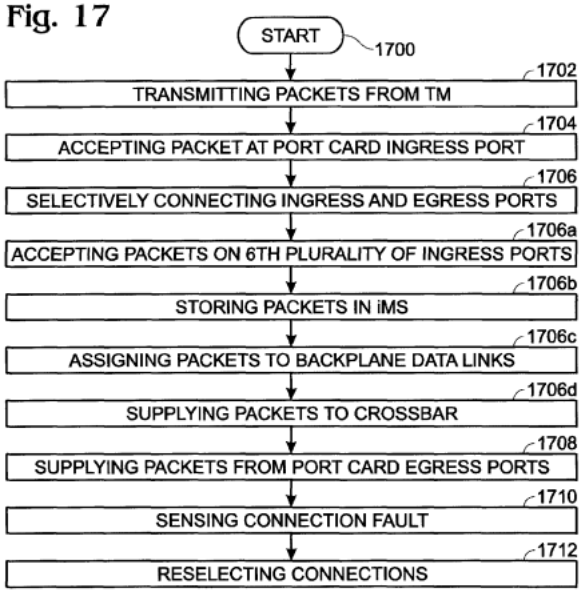
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

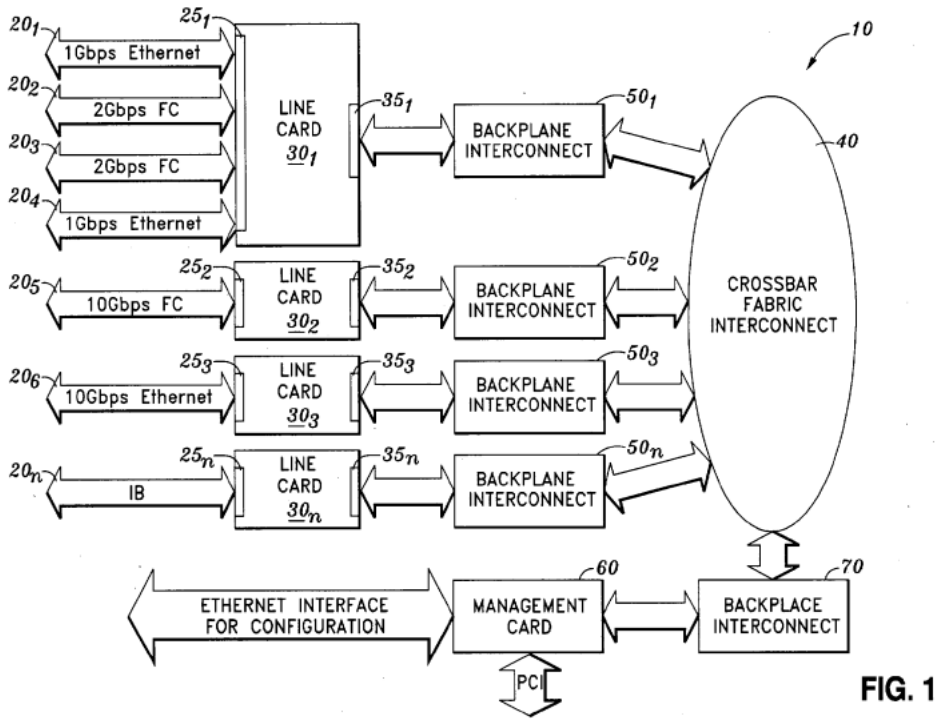
| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 18 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |

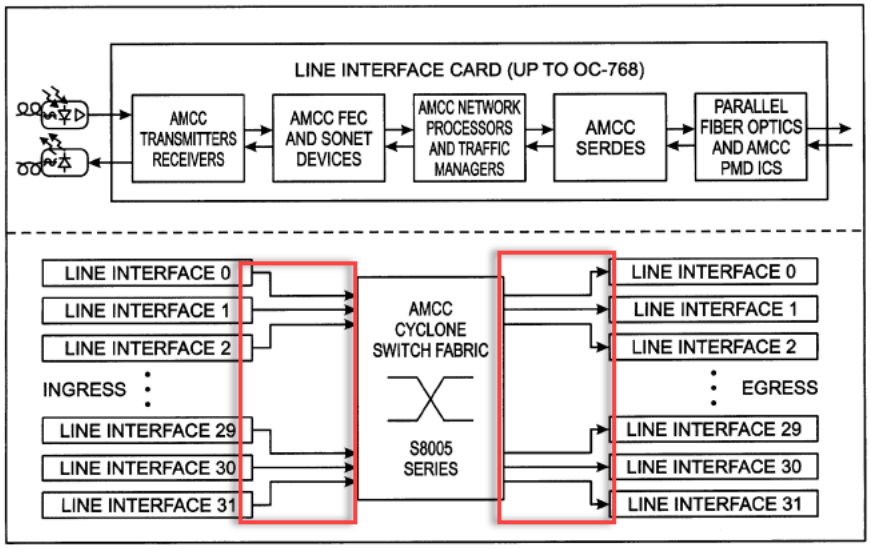
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| 19[preamble] | Apparatus for connecting a network node with a communication network, comprising: | The Reference discloses apparatus for connecting a network node with a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 19[a] | one or more interface modules, which are arranged to process data frames having frame attributes sent | The Reference discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. |

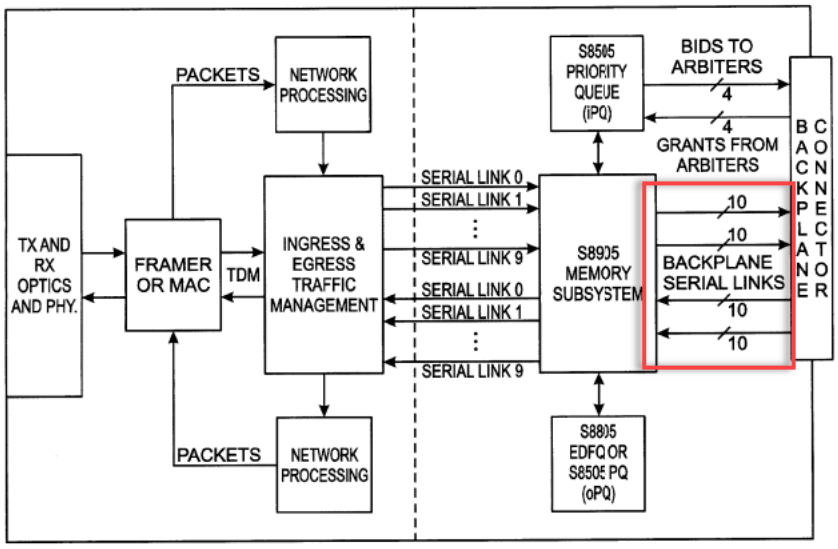
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | between the network node and the communication network; | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 19[b] | a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules; | The Reference discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 19[c] | a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and | The Reference discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu. |

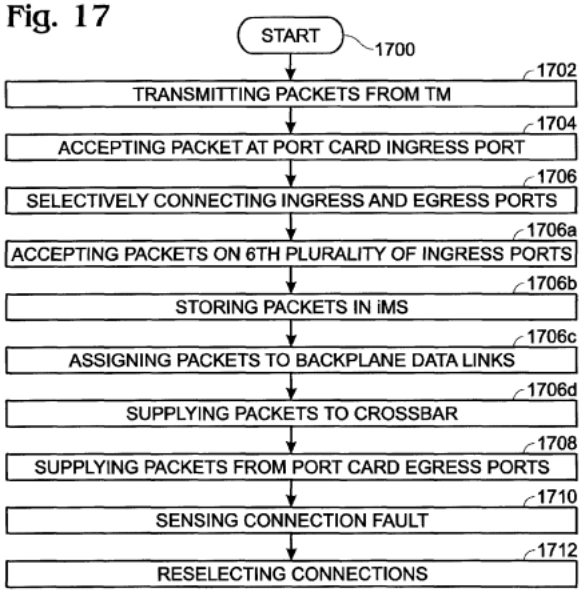
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

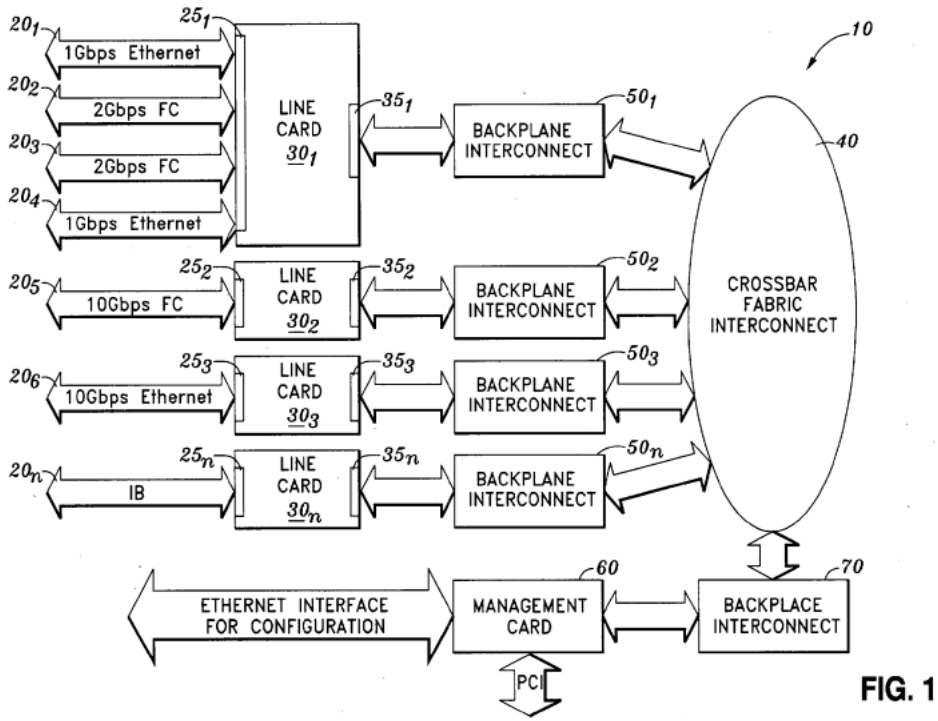
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

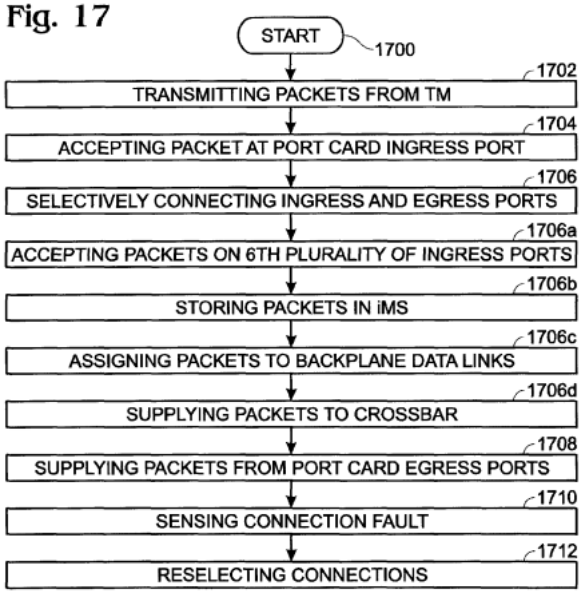
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 19[d] | a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame, | The Reference discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown |

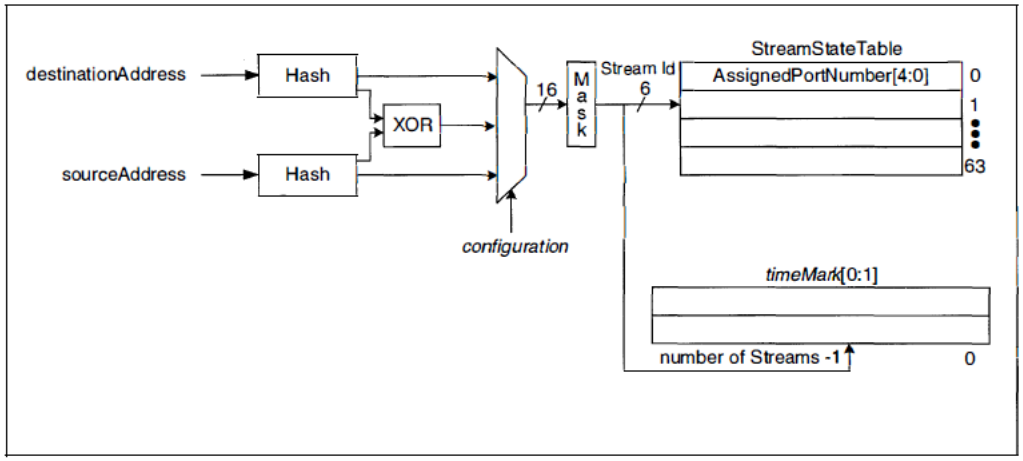
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

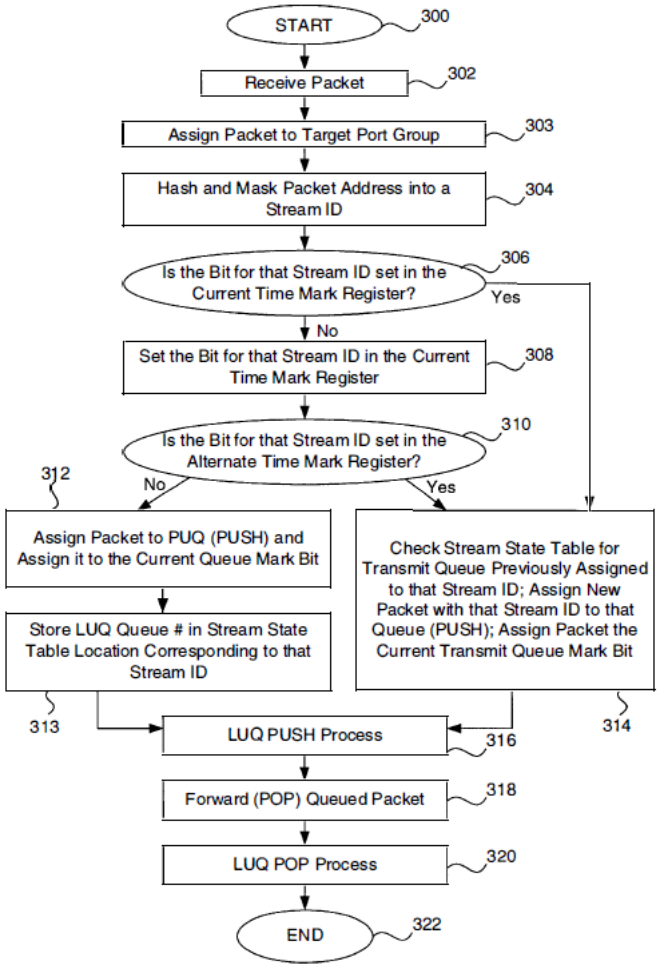
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | • Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br><br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

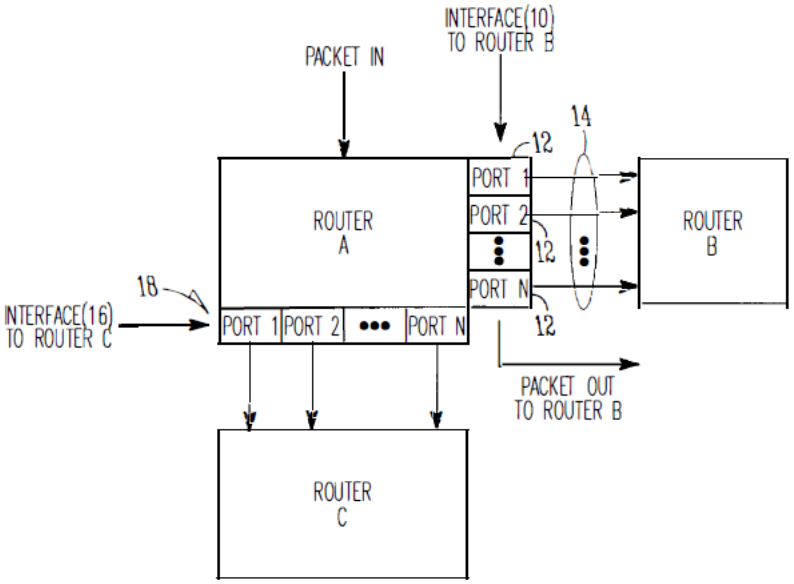
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

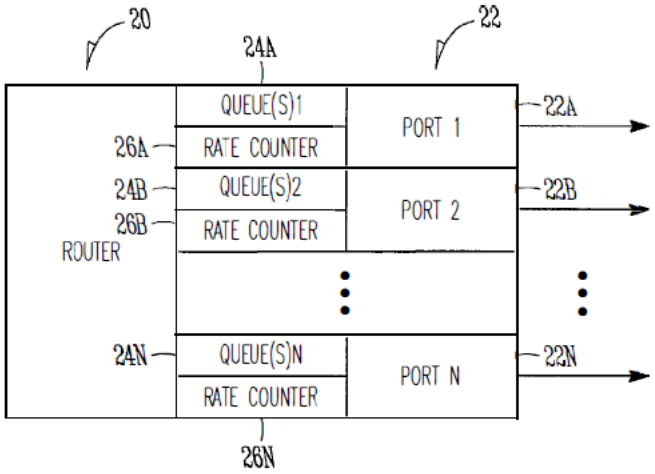
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

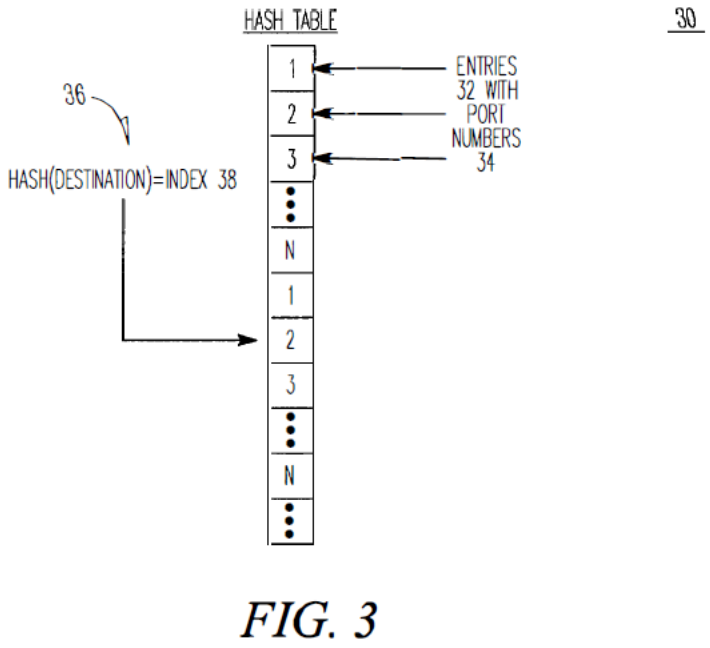
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

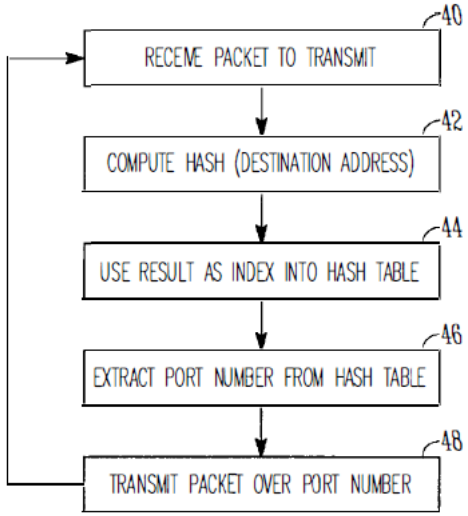
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

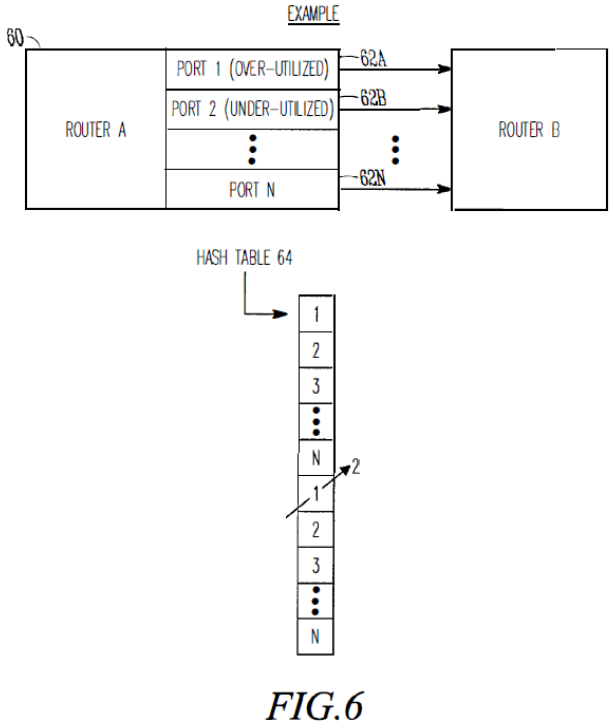
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

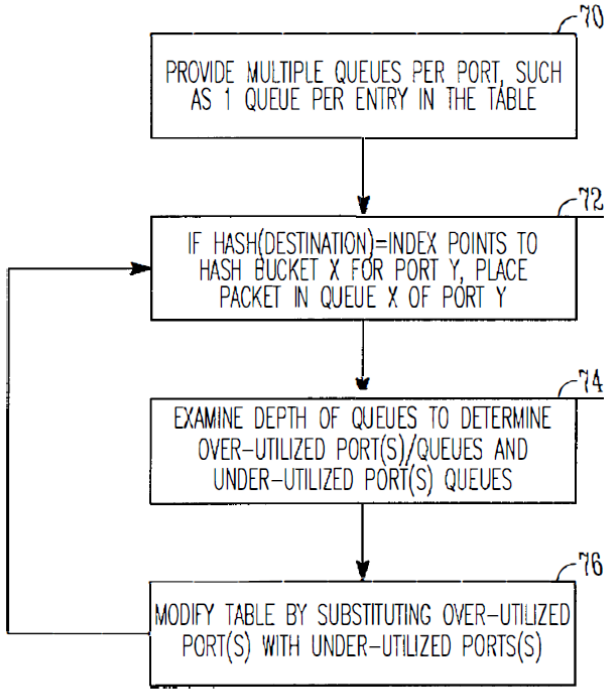
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

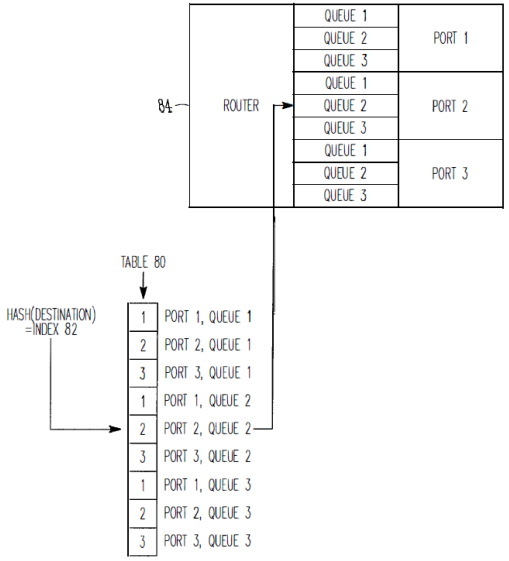
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

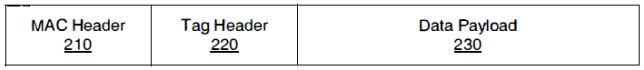
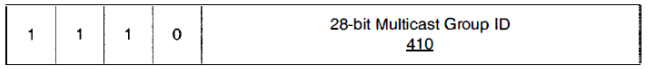
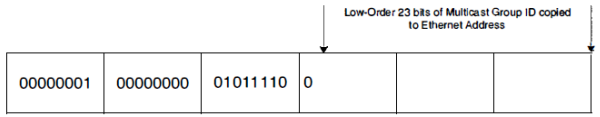
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by 548ubstitute- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

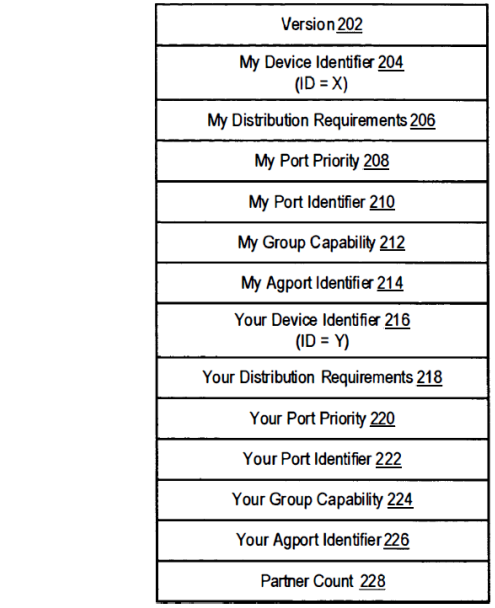
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| 19[e] | at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. | The Reference discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below is an example.<br><br>Smith '430 at 5:51-64 ("The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be oper-ated as a single logical link, referred to herein as a virtual link bundle. Network device 120(1) operates the two links cou-pling network device 120(1) to virtual network device 202 as 55 a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corre-sponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 60 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, virtual link bundles 250(1) and 250(2) are each operated as an EtherChannel™ or as an aggregated link (as described in IEEE 802.3).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the link aggregation technique. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914 |

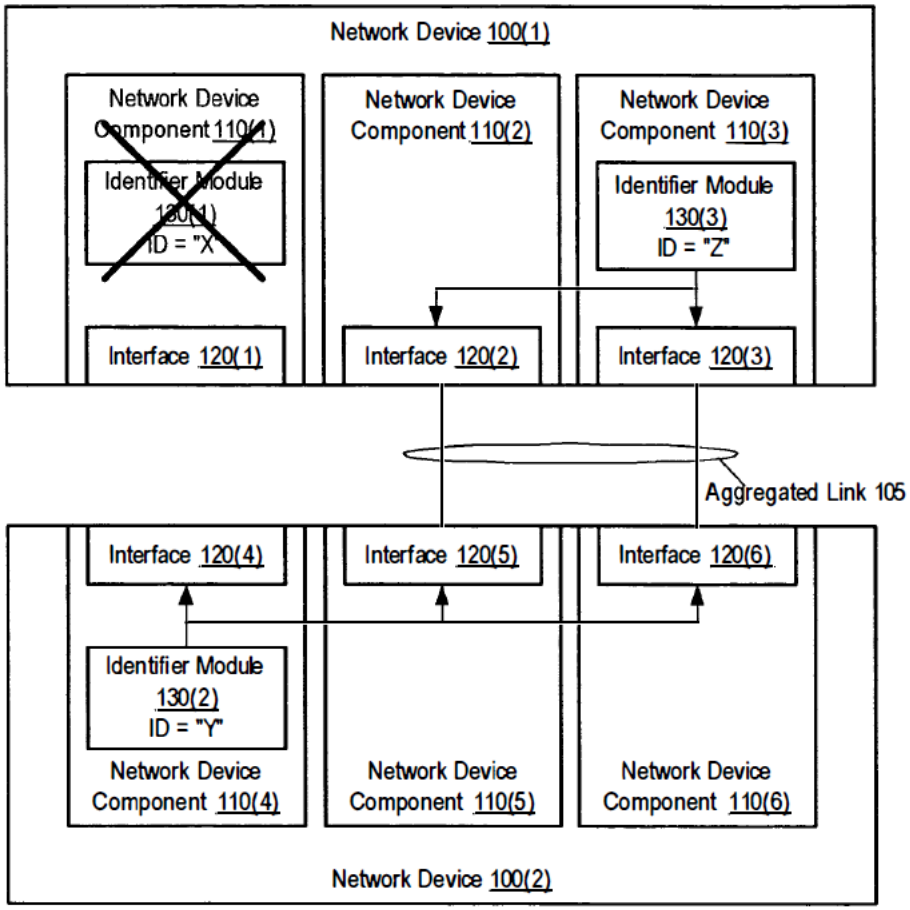
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | • Borgione '125<br><br>DeJager '424 at Abstract ("Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.")<br><br>DeJager '424 at 1:38-50 ("One way to relieve this bottle-neck is to provide a logical grouping of multiple ports into a single port. The bandwidth of the new port is increased since it has multiple lines (cables) connecting a switch and another network device, each line capable of carrying data at the same rate as the line connecting data sources to the switch. This grouping of ports is sometimes referred to as a port aggregation or port group. One example of such a port aggregation implementation is Cisco Technology, Inc.'s Fast EtherChannel™ port group in a Fast Ethernet network. Further information regarding Fast EtherChannel™ may be found on Cisco Technology, Inc.'s World Wide Web site www.cisco.com. This information is incorporated by reference herein for all purposes.")<br><br>DeJager '424 at 2:47-65 ("The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current |

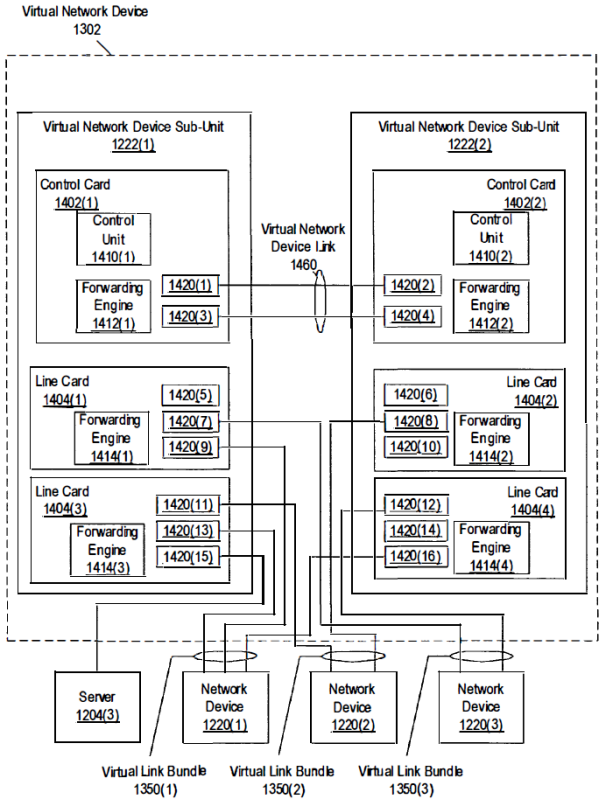
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.")<br><br>DeJager '424 at 2:67-3:15 ("In one aspect, the present invention provides a method of distributing traffic over a network port group. The method involves receiving a packet of data to be forwarded, deter-mining a stream ID for the packet, and determining whether a prior packet having that stream ID has been distributed to a queue on a port in the group during a predetermined time interval. Where a prior packet having that stream ID has not been distributed to a queue on a port of the group during the predetermined time interval, the method involves allocating the packet to a queue of a port having a lesser load in its queue than a queue of any other port of the group. The method may also involve, where a prior packet having that stream ID has been distributed to a queue on a port of the group during the predetermined time interval, allocating the packet to that queue. In addition, the method may involve monitoring the port group queues to maintain proper iden-tification of the least utilized queue.")<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 4:29-46 ("The present invention provides methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present inven-tion is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other network environments.")<br><br>DeJager '424 at 4:47-58 ("FIG. 1 illustrates a block diagram of a simple network. The network 100 includes two servers S1. and S2, respectively, and two switches, X1 and X2, respectively, as well as four clients C1, C2, C3 and C4, respectively. Clients C , , and C4 are connected to switch X1 by, for example, Fast Ethernet links 102 via ports 1, 2, 3 and 4, respectively. Server S1 is connected to switch X1 via a port aggregation 104, which is a port group composed of ports 5 and 6 of switch X1 . Switch X1 is connected to switch X2 via a second port aggregation 106 which includes ports 7, 8 and 9. Switch X2 is connected to server S2 via port O and Fast Ethernet link 108.")<br><br>Dontu at Abstract ("Various methods and systems for preventing erroneous link aggregation due to component relocation are disclosed. Such methods include a method for changing the identifier used by a network device and communicating the identifier change to a peer |

| No. | ’740 Patent Claim 19 | The Reference |
|---|---|---|
| | | network device without disrupting an aggregated link. In one embodiment, a method involves detecting an identifier change and sending a Port Aggrega-tion Protocol (PAgP) protocol data unit (PDU) that includes a new identifier and information. The information indicates the identifier change. The new identifier identifies a network device subsequent to the identifier change. Another embodi-ment of a method involves detecting an identifier change and, subsequent to the identifier change, sending a link aggregation protocol PDU that includes an "old device identifier" field dedicated to conveying an old identifier. The old identifier identifies a network device prior to the iden-tifier change.”)<br><br>Dontu at Figure 2 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Dontu at Figure 3 |

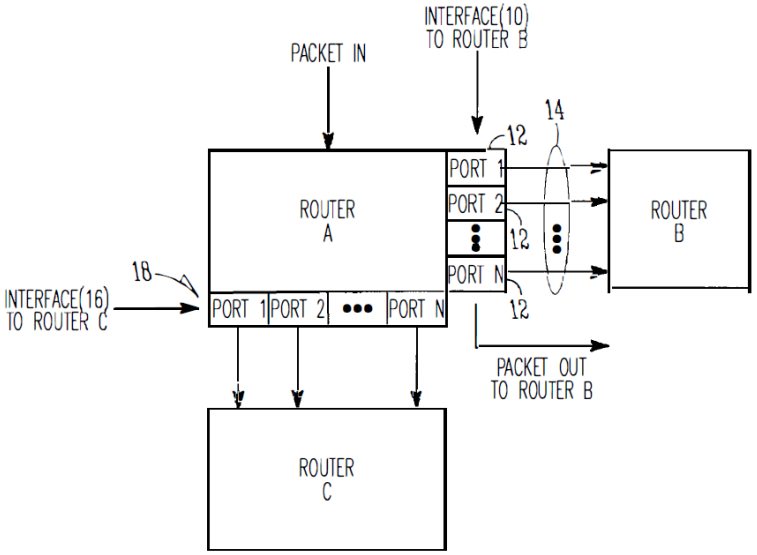
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Dontu at Figure 14 |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | |  <br>FIG. 14<br><br>Dontu at [0004] ("Link aggregation is used to logically combine two or more individual links into a single aggregated link. Link aggregation can provide improved performance and increased fault tolerance. Improved performance arises because the aggregated link appears to have a bandwidth equal to the combined bandwidth of the individual links. Traffic can be load-balanced among the individual links. Increased fault tolerance is provided since one or more individual links within an aggregated link can fail without disrupting communication between the devices coupled by the aggregated link. Link aggregation techniques include |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Link Aggregation Control Protocol (LACP), which is defined in IEEE 803.2ad, and Port Aggregation Protocol (PAgP), which is a standard promulgated by CISCO SYS-TEMS, INC.")<br><br>Dontu at [0012] ("The method can also involve detecting whether a partner interface is executing a compatible version of PAgP. If the partner interface is not executing the compatible version of PAgP, the compatible version of PAgP can be provided to the partner interface. Alternatively, if the partner interface is not executing the compatible version of PAgP, the partner interface can be inhibited from including a link in an aggregated link.")<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).") |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Dontu at [0037] ("In this example, the network devices 100(1) and 100(2) use Port Aggregation Protocol (PAgP) to form aggre-gated links. Network devices 100(1) each send PAgP pro-tocol data units (PDUs) to each other in order to determine whether any of the links between the two network devices can be combined into an aggregated link. Each PAgP PDU includes an identifier that uniquely identifies the network device that sent that PAgP PDU. Within network device 100(1), identifier module 130(1) of network device compo-nent 110(1) supplies an identifier "X" to each of the inter-faces 120(1)-120(3) within network device 100(1). Inter-faces 120(1)-120(3) include identifier X in each PAgP PDU sent by those interfaces. Similarly, identifier module 130(2) of network device component 110( 4) supplies an identifier "Y" to each interface 120( 4)-120( 6) of network device 100(2). Interfaces 120( 4)-120( 6) include identifier Yin each PAgP PDU sent by those interfaces.")<br><br>Dontu at [0040] ("FIG. 2 illustrates some of the fields that can be included in a PAgP PDU. As shown, PDU 200 includes Version field 202, My Device Identifier field 204 ("My" refers to the device sending the PAgP PDU), My Distribu-tion Requirements field 206, My Port Priority field 208, My Port Identifier field 212, My Group Capability field 212, My Agport (Aggregated Port) Identifier field 214, Your Device Identifier field 216 ("Your" refers to the device to which the PAgP PDU is being sent), Your Distribution Requirements field 218, Your Port Priority field 220, Your Port Identifier field 222, Your Group Capability field 224, Your Agport Identifier field 226, and Partner Count field 228.")<br><br>Dontu at [0110] ("Interfaces 1420(13), 1420(9), and 1420(16), which are each coupled to network device 1220(1) by virtual link bundle 1350(1), form an interface bundle (e.g., an Ether-Channel (TM) port bundle). Similarly, interfaces 1420(11) and 1420(8) form another interface bundle that is coupled to network device 1220(2) by virtual link bundle 1350(2). Interfaces 1420(7) and 1420(12) form a third interface bundle that is coupled to network device 1220(3) by virtual link bundle 1350(3). Within virtual network device 1302, each interface in the same interface bundle is assigned the same logical identifier. For example, interfaces 1420(13), 1420(9), and 1420(16) are each assigned the same logical identifier. In some embodiments, packets received via one of these interfaces are tagged or otherwise associated with the logical identifier to indicate that those packets were received via the |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | virtual link bundle coupling virtual network device 1302 to network device 1220(1). It is noted that similar interface bundles are implemented within each network device 1220(1)-1220(3), and that interfaces included in such bundles are also assigned the same logical identifier by each network device ( or by virtual network device 1302, in embodiments in which virtual network device 1302 controls the configuration of the network devices 1220(1)-1220(3)). For example, network device 1220(1) can assign the same logical identifier to each of the interfaces coupled to virtual link bundle 1350(1).")<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 2:6-22 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.") |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Li '914 at 4:9-25 ("Referring to FIG. 1, a Router A is shown having an inter-face 10 with a plurality of ports which connect Router A with Router B over a plurality of connections, lines, wires, links or bundled links 14. The ports 12 of Router A are configured to permit transmission of packets from Router A to Router B, and these ports 12 may be referred to as output ports, egress ports, links, or the like. As shown in FIG. 1, port 1 to port N may be connected with Router B, and there may be additional interfaces 16 having ports 18 connected with other routers in the network. When a packet is received by Router A, Router A determines whether the received packet should be transmitted to Router B or to other routers connected to Router A, based in part upon the destination address of the packet. If a packet is to be transmitted from Router A to Router B, then Router A may transmit this packet over the one of the ports 12 shown in FIG. 1.")<br><br>Li '914 at Figure 1<br>  <br><br>*FIG. 1* |

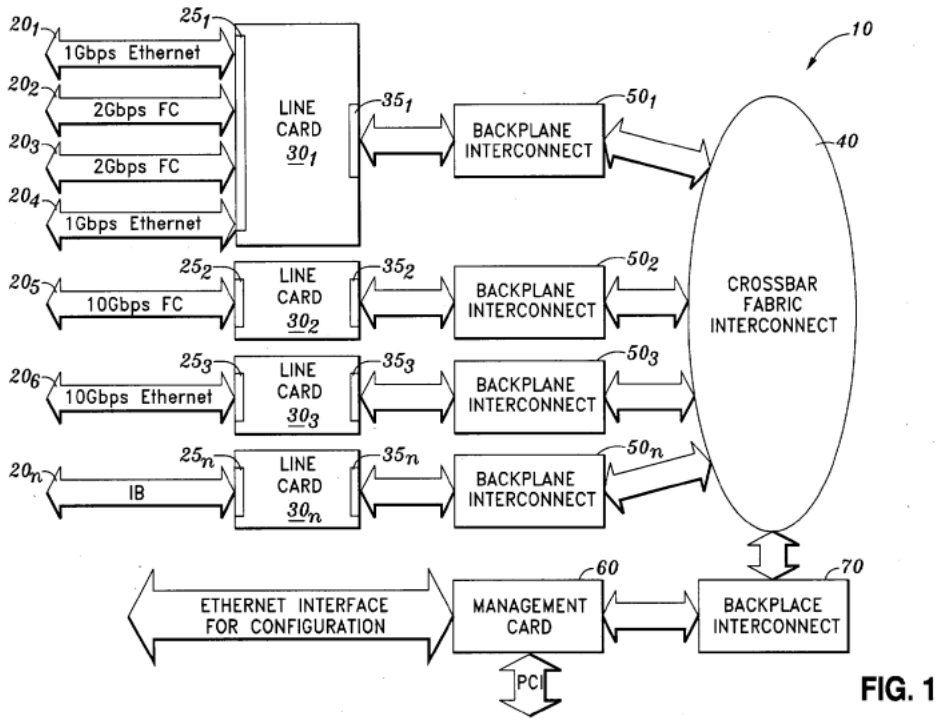
| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | Borgione '125 at 1:55-65 ("Link nodes 110 and 120 can be in physically remote locations, thereby connecting their associated local area networks (LANs). The plurality of network links 150 between link nodes 110 and 120 can be aggregated as a single logical link over which all traffic between link nodes 110 and 120 is distributed. Such aggregation multiplies the available bandwidth for communications between link nodes 110 and 120, and therefore between the two local area networks. When appropriately configured, such a connection can permit the two local area networks to interact as if they were one large local area network.")<br><br>Borgione '125 at 1:66-2:7 ("As stated above, the plurality of network links between 110 and 120 can be aggregated as a single logical link. In this manner, each link node 110 and 120 sees the plurality of network links between them as one logical interface. One type of such an aggregate of links is an EtherChannel, a protocol that allows up to eight Fast Ethernet or Gigabit Ethernet links to be aggregated. Routing protocols treat the aggregated links as a single, routed interface with a common IP address.")<br><br>Borgione '125 at 5:28-50 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even |

| No. | '740 Patent Claim 19 | The Reference |
|---|---|---|
| | | distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.") |

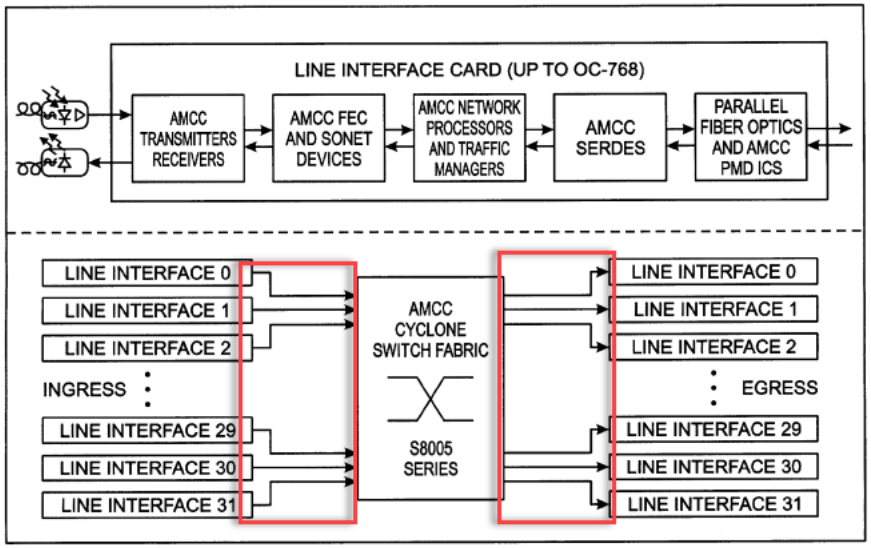
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| 20[preamble] | Apparatus for connecting a network node with a communication network, comprising: | The Reference discloses apparatus for connecting a network node with a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 20[a] | one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network; | The Reference discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

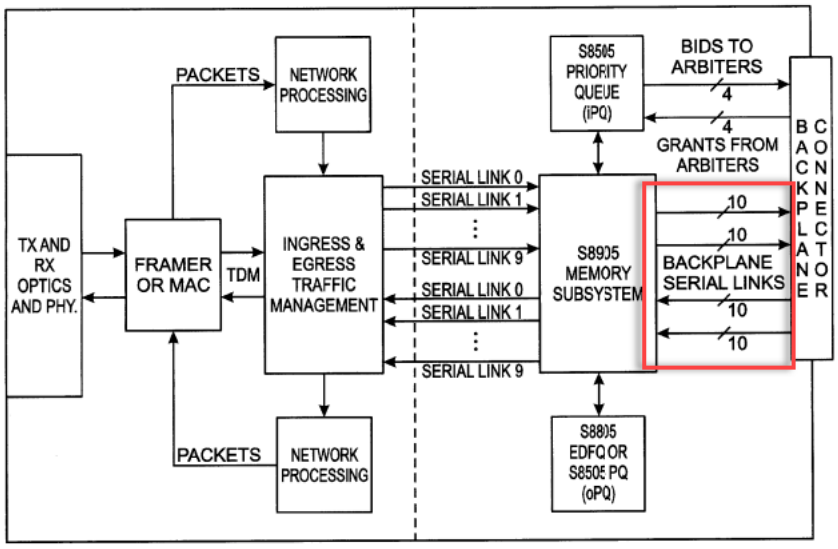
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| 20[b] | a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules; | The Reference discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 20[c] | a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and | The Reference discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards |

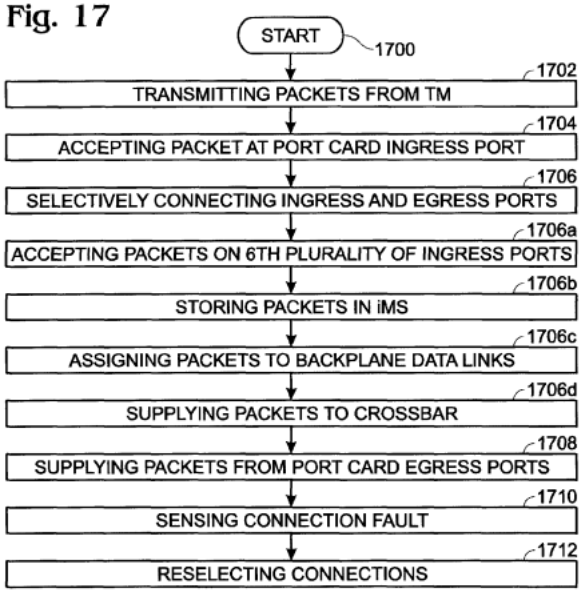
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

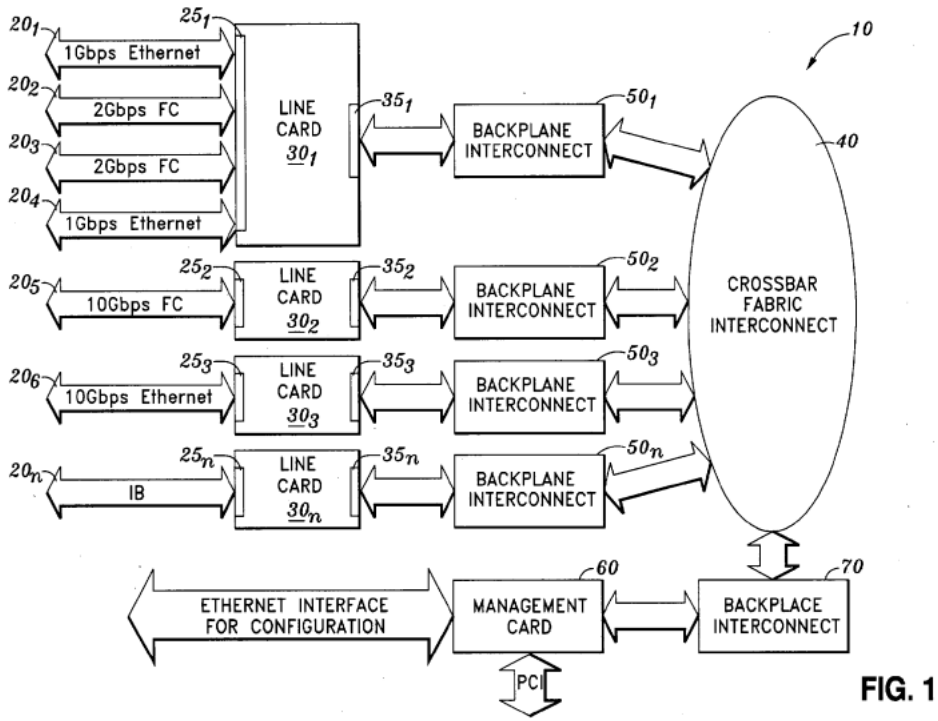
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

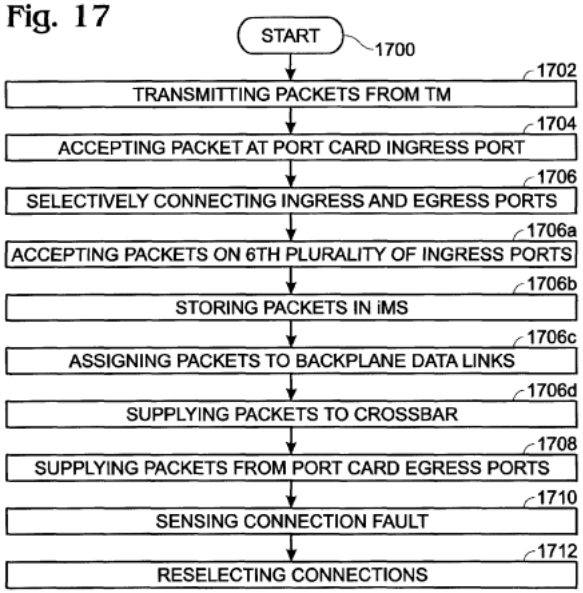
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 20[d] | a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame, | The Reference discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces |

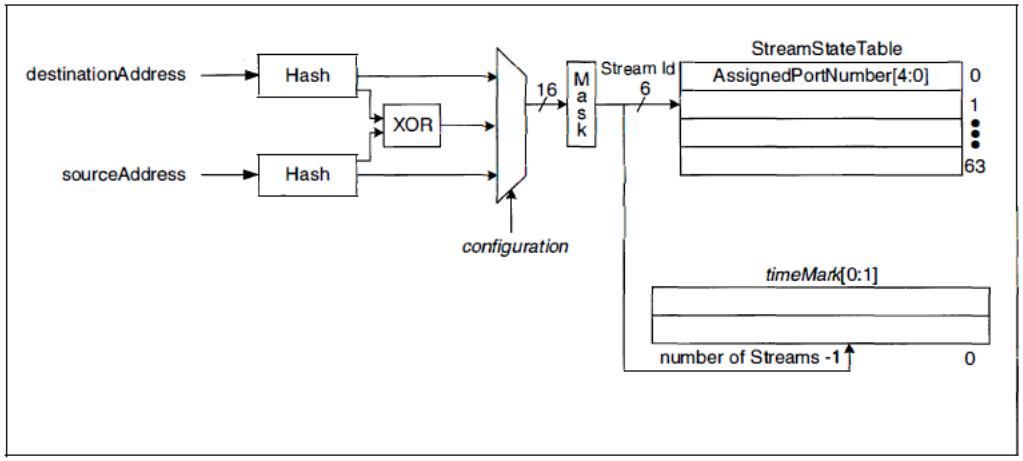
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

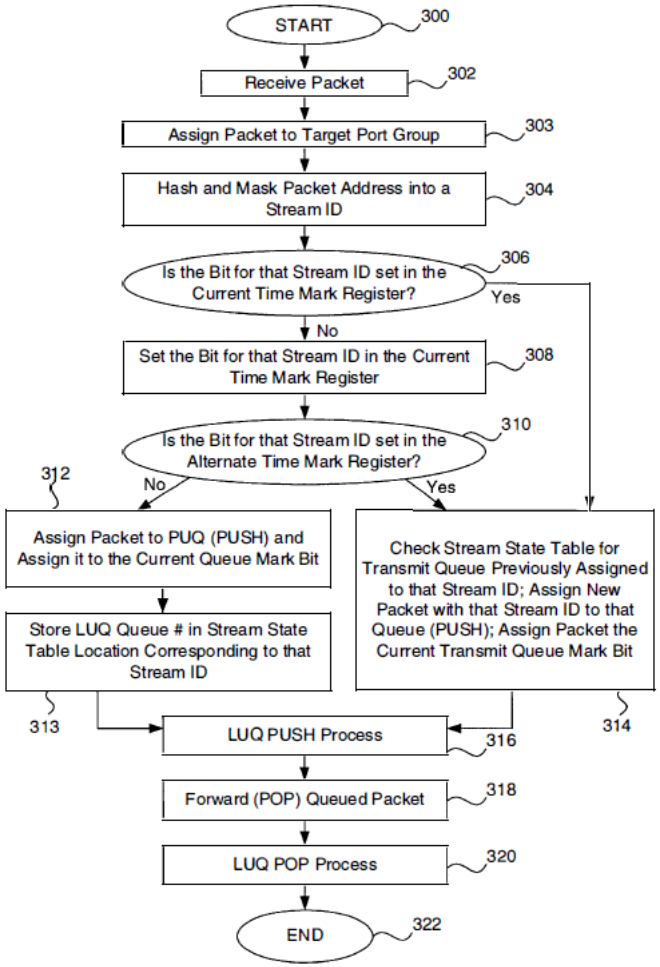
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | • Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

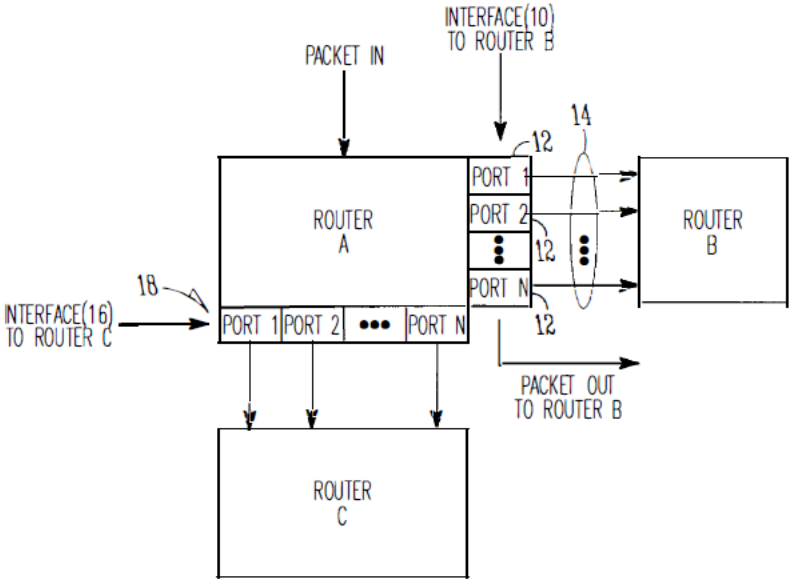
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

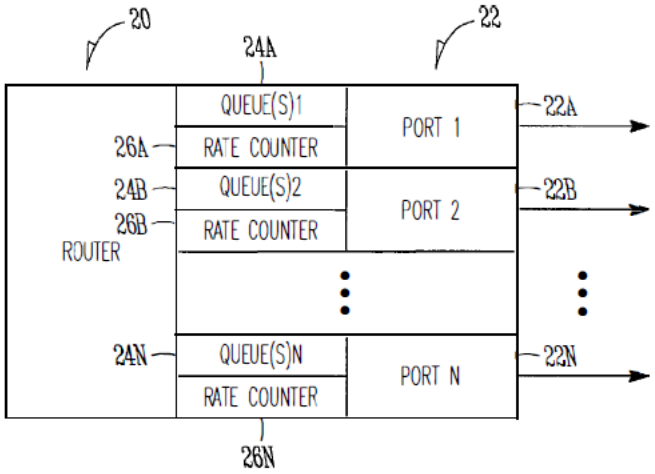
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

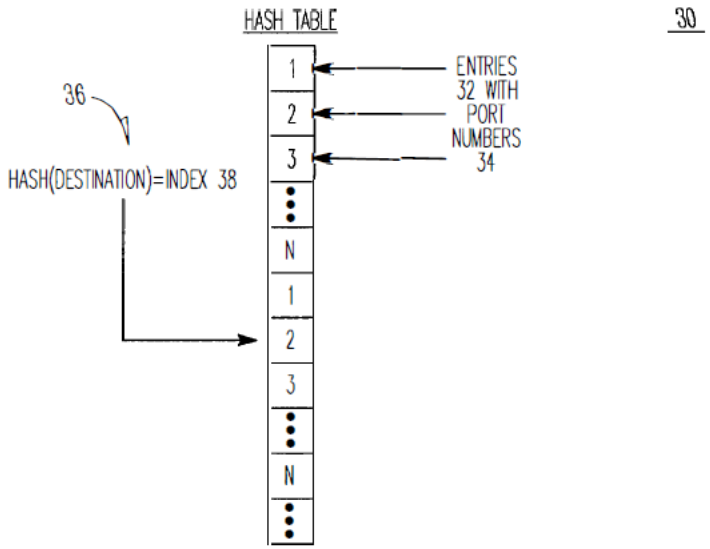
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

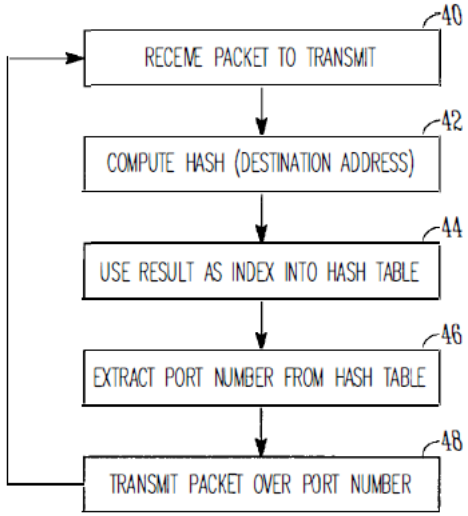
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

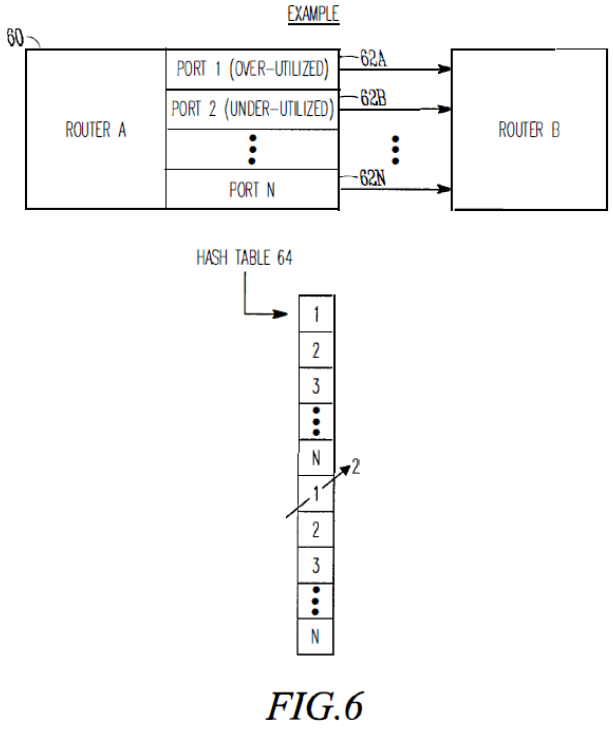
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

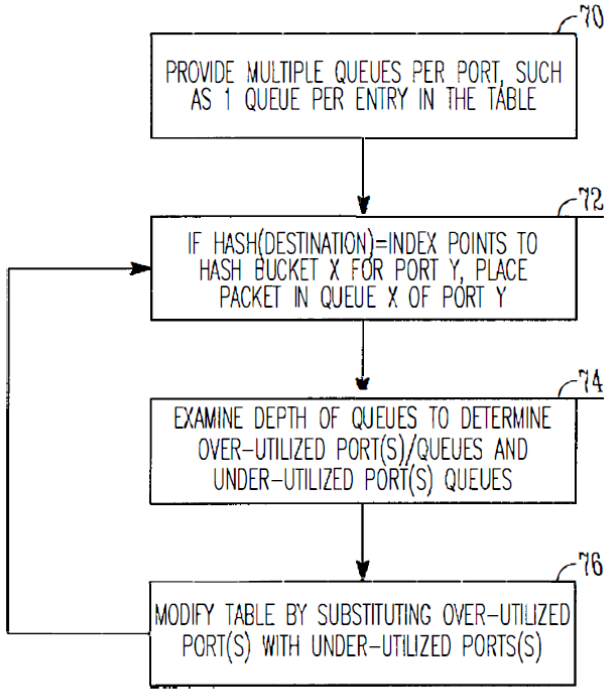
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

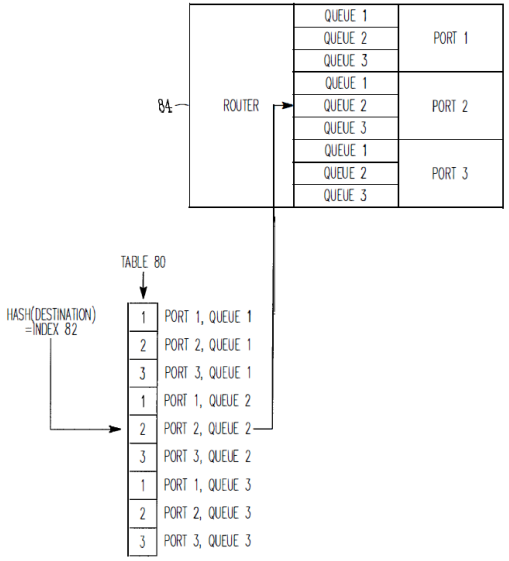
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

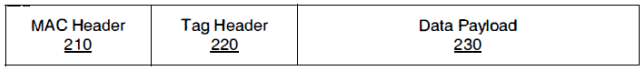
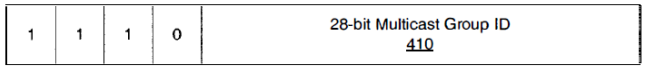
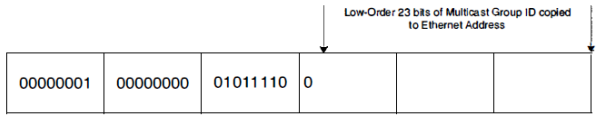
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

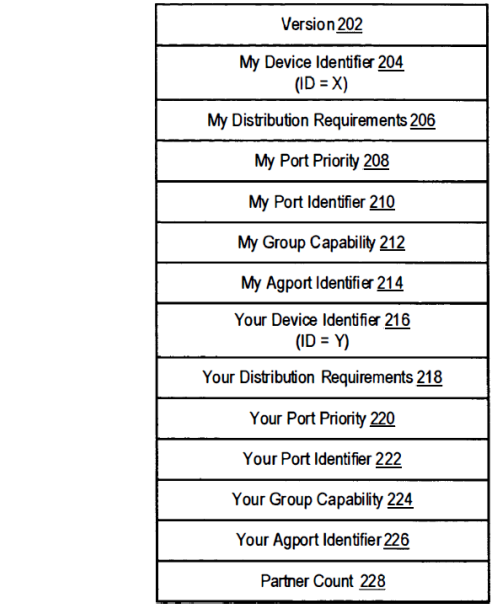
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| 20[e] | two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. | The Reference discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below is an example.<br><br>Smith '430 at 5:51-64 ("The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be oper-ated as a single logical link, referred to herein as a virtual link bundle. Network device 120(1) operates the two links cou-pling network device 120(1) to virtual network device 202 as 55 a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corre-sponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 60 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, virtual link bundles 250(1) and 250(2) are each operated as an EtherChannel™ or as an aggregated link (as described in IEEE 802.3).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the link aggregation technique. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

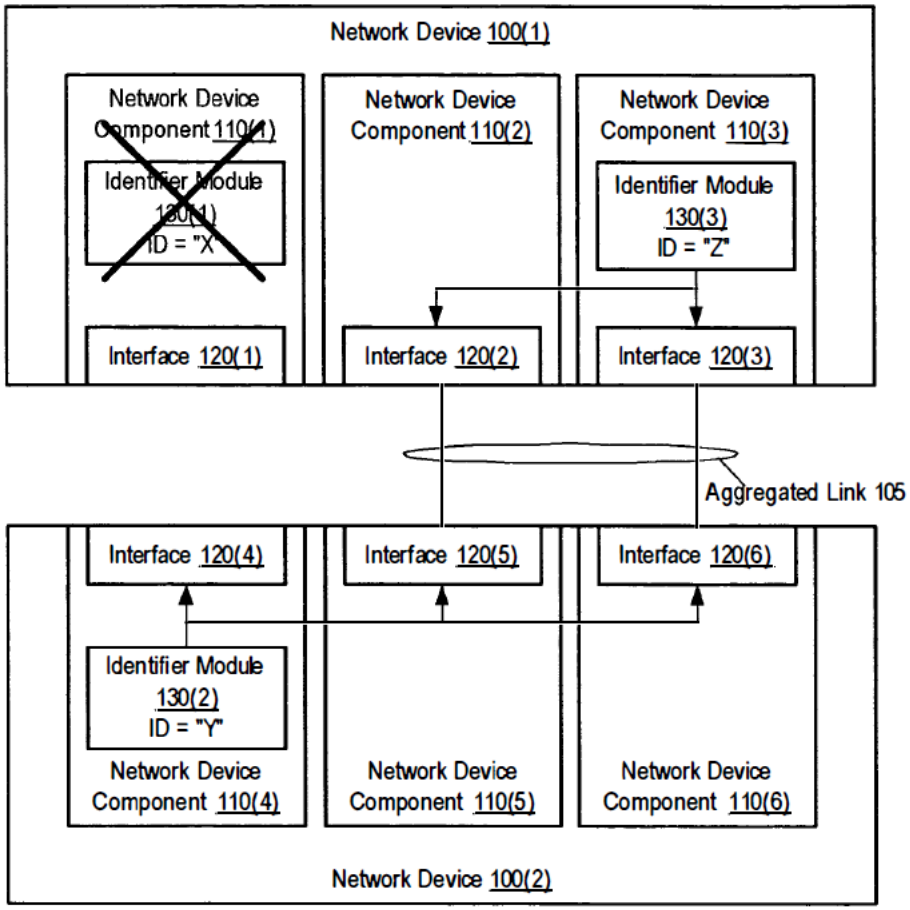
| No. | ’740 Patent Claim 20 | The Reference |
|---|---|---|
| | | • Li ’914<br>• Borgione ’125<br><br>DeJager ’424 at Abstract (“Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.”)<br><br>DeJager ’424 at 1:38-50 (“One way to relieve this bottle-neck is to provide a logical grouping of multiple ports into a single port. The bandwidth of the new port is increased since it has multiple lines (cables) connecting a switch and another network device, each line capable of carrying data at the same rate as the line connecting data sources to the switch. This grouping of ports is sometimes referred to as a port aggregation or port group. One example of such a port aggregation implementation is Cisco Technology, Inc.'s Fast EtherChannel™ port group in a Fast Ethernet network. Further information regarding Fast EtherChannel™ may be found on Cisco Technology, Inc.'s World Wide Web site www.cisco.com. This information is incorporated by reference herein for all purposes.”)<br><br>DeJager ’424 at 2:47-65 (“The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets |

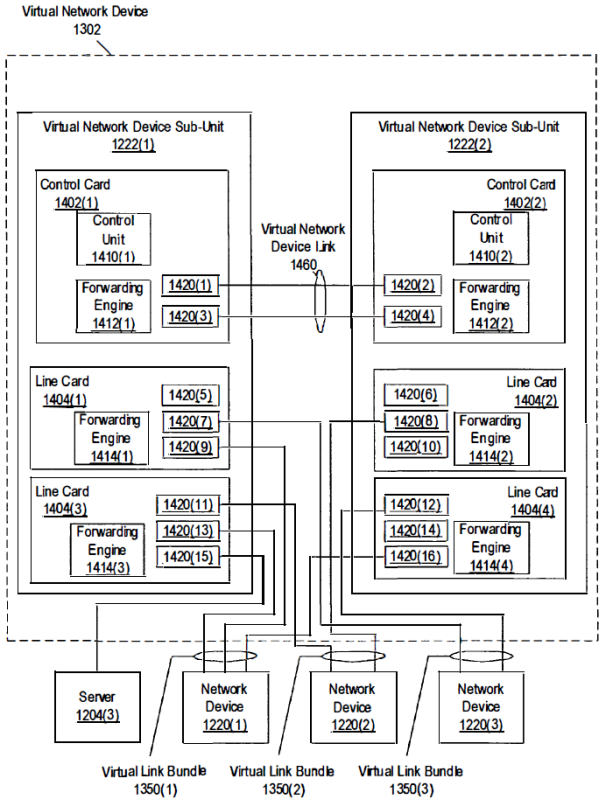
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | from a given stream may be forwarded on different ports depending upon each port's current utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.")<br><br>DeJager '424 at 2:67-3:15 ("In one aspect, the present invention provides a method of distributing traffic over a network port group. The method involves receiving a packet of data to be forwarded, deter-mining a stream ID for the packet, and determining whether a prior packet having that stream ID has been distributed to a queue on a port in the group during a predetermined time interval. Where a prior packet having that stream ID has not been distributed to a queue on a port of the group during the predetermined time interval, the method involves allocating the packet to a queue of a port having a lesser load in its queue than a queue of any other port of the group. The method may also involve, where a prior packet having that stream ID has been distributed to a queue on a port of the group during the predetermined time interval, allocating the packet to that queue. In addition, the method may involve monitoring the port group queues to maintain proper iden-tification of the least utilized queue.")<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 4:29-46 ("The present invention provides methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present inven-tion is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other network environments.")<br><br>DeJager '424 at 4:47-58 ("FIG. 1 illustrates a block diagram of a simple network. The network 100 includes two servers S1. and S2, respectively, and two switches, X1 and X2, respectively, as well as four clients C1, C2, C3 and C4, respectively. Clients C , , and C4 are connected to switch X1 by, for example, Fast Ethernet links 102 via ports 1, 2, 3 and 4, respectively. Server S1 is connected to switch X1 via a port aggregation 104, which is a port group composed of ports 5 and 6 of switch X1 . Switch X1 is connected to switch X2 via a second port aggregation 106 which includes ports 7, 8 and 9. Switch X2 is connected to server S2 via port O and Fast Ethernet link 108.")<br><br>Dontu at Abstract ("Various methods and systems for preventing erroneous link aggregation due to component relocation are disclosed. Such methods include a method for changing the |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | identifier used by a network device and communicating the identifier change to a peer network device without disrupting an aggregated link. In one embodiment, a method involves detecting an identifier change and sending a Port Aggrega-tion Protocol (PAgP) protocol data unit (PDU) that includes a new identifier and information. The information indicates the identifier change. The new identifier identifies a network device subsequent to the identifier change. Another embodi-ment of a method involves detecting an identifier change and, subsequent to the identifier change, sending a link aggregation protocol PDU that includes an "old device identifier" field dedicated to conveying an old identifier. The old identifier identifies a network device prior to the iden-tifier change.")<br><br>Dontu at Figure 2 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Dontu at Figure 3 |

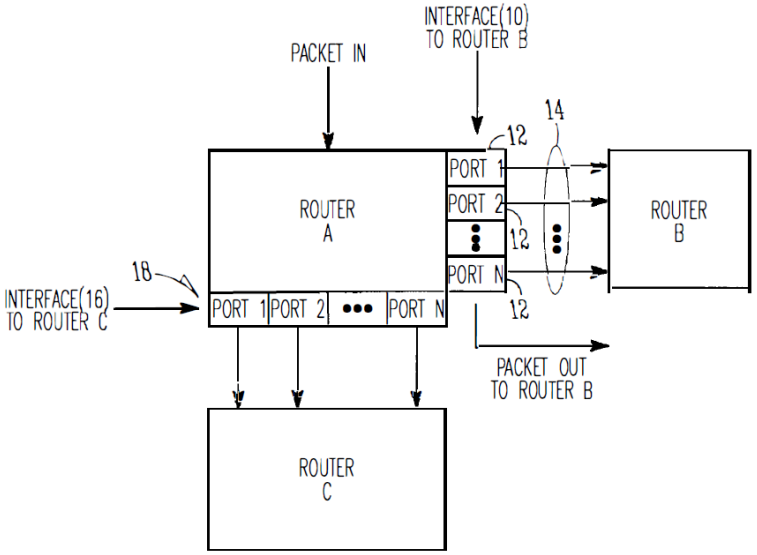
| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Dontu at Figure 14 |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | |  <br>FIG. 14<br><br>Dontu at [0004] ("Link aggregation is used to logically combine two or more individual links into a single aggregated link. Link aggregation can provide improved performance and increased fault tolerance. Improved performance arises because the aggregated link appears to have a bandwidth equal to the combined bandwidth of the individual links. Traffic can be load-balanced among the individual links. Increased fault tolerance is provided since one or more individual links within an aggregated link can fail without disrupting communication between the devices coupled by the aggregated link. Link aggregation techniques include |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Link Aggregation Control Protocol (LACP), which is defined in IEEE 803.2ad, and Port Aggregation Protocol (PAgP), which is a standard promulgated by CISCO SYS-TEMS, INC.")<br><br>Dontu at [0012] ("The method can also involve detecting whether a partner interface is executing a compatible version of PAgP. If the partner interface is not executing the compatible version of PAgP, the compatible version of PAgP can be provided to the partner interface. Alternatively, if the partner interface is not executing the compatible version of PAgP, the partner interface can be inhibited from including a link in an aggregated link.")<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).") |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Dontu at [0037] ("In this example, the network devices 100(1) and 100(2) use Port Aggregation Protocol (PAgP) to form aggre-gated links. Network devices 100(1) each send PAgP pro-tocol data units (PDUs) to each other in order to determine whether any of the links between the two network devices can be combined into an aggregated link. Each PAgP PDU includes an identifier that uniquely identifies the network device that sent that PAgP PDU. Within network device 100(1), identifier module 130(1) of network device compo-nent 110(1) supplies an identifier "X" to each of the inter-faces 120(1)-120(3) within network device 100(1). Inter-faces 120(1)-120(3) include identifier X in each PAgP PDU sent by those interfaces. Similarly, identifier module 130(2) of network device component 110( 4) supplies an identifier "Y" to each interface 120( 4)-120( 6) of network device 100(2). Interfaces 120( 4)-120( 6) include identifier Yin each PAgP PDU sent by those interfaces.")<br><br>Dontu at [0040] ("FIG. 2 illustrates some of the fields that can be included in a PAgP PDU. As shown, PDU 200 includes Version field 202, My Device Identifier field 204 ("My" refers to the device sending the PAgP PDU), My Distribu-tion Requirements field 206, My Port Priority field 208, My Port Identifier field 212, My Group Capability field 212, My Agport (Aggregated Port) Identifier field 214, Your Device Identifier field 216 ("Your" refers to the device to which the PAgP PDU is being sent), Your Distribution Requirements field 218, Your Port Priority field 220, Your Port Identifier field 222, Your Group Capability field 224, Your Agport Identifier field 226, and Partner Count field 228.")<br><br>Dontu at [0110] ("Interfaces 1420(13), 1420(9), and 1420(16), which are each coupled to network device 1220(1) by virtual link bundle 1350(1), form an interface bundle (e.g., an Ether-Channel (TM) port bundle). Similarly, interfaces 1420(11) and 1420(8) form another interface bundle that is coupled to network device 1220(2) by virtual link bundle 1350(2). Interfaces 1420(7) and 1420(12) form a third interface bundle that is coupled to network device 1220(3) by virtual link bundle 1350(3). Within virtual network device 1302, each interface in the same interface bundle is assigned the same logical identifier. For example, interfaces 1420(13), 1420(9), and 1420(16) are each assigned the same logical identifier. In some embodiments, packets received via one of these interfaces are tagged or otherwise associated with the logical identifier to indicate that those packets were received via the |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | virtual link bundle coupling virtual network device 1302 to network device 1220(1). It is noted that similar interface bundles are implemented within each network device 1220(1)-1220(3), and that interfaces included in such bundles are also assigned the same logical identifier by each network device ( or by virtual network device 1302, in embodiments in which virtual network device 1302 controls the configuration of the network devices 1220(1)-1220(3)). For example, network device 1220(1) can assign the same logical identifier to each of the interfaces coupled to virtual link bundle 1350(1).")<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 2:6-22 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.") |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Li '914 at 4:9-25 ("Referring to FIG. 1, a Router A is shown having an inter-face 10 with a plurality of ports which connect Router A with Router B over a plurality of connections, lines, wires, links or bundled links 14. The ports 12 of Router A are configured to permit transmission of packets from Router A to Router B, and these ports 12 may be referred to as output ports, egress ports, links, or the like. As shown in FIG. 1, port 1 to port N may be connected with Router B, and there may be additional interfaces 16 having ports 18 connected with other routers in the network. When a packet is received by Router A, Router A determines whether the received packet should be transmitted to Router B or to other routers connected to Router A, based in part upon the destination address of the packet. If a packet is to be transmitted from Router A to Router B, then Router A may transmit this packet over the one of the ports 12 shown in FIG. 1.")<br><br>Li '914 at Figure 1<br>  <br><br>*FIG. 1* |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | Borgione '125 at 1:55-65 ("Link nodes 110 and 120 can be in physically remote locations, thereby connecting their associated local area networks (LANs). The plurality of network links 150 between link nodes 110 and 120 can be aggregated as a single logical link over which all traffic between link nodes 110 and 120 is distributed. Such aggregation multiplies the available bandwidth for communications between link nodes 110 and 120, and therefore between the two local area networks. When appropriately configured, such a connection can permit the two local area networks to interact as if they were one large local area network.")<br><br>Borgione '125 at 1:66-2:7 ("As stated above, the plurality of network links between 110 and 120 can be aggregated as a single logical link. In this manner, each link node 110 and 120 sees the plurality of network links between them as one logical interface. One type of such an aggregate of links is an EtherChannel, a protocol that allows up to eight Fast Ethernet or Gigabit Ethernet links to be aggregated. Routing protocols treat the aggregated links as a single, routed interface with a common IP address.")<br><br>Borgione '125 at 5:28-50 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even |

| No. | '740 Patent Claim 20 | The Reference |
|---|---|---|
| | | distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.") |

| No. | '740 Patent Claim 21 | The Reference |
|---|---|---|
| 21 | The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. | The Reference discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

| No. | '740 Patent Claim 22 | The Reference |
|---|---|---|
| 22 | The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. | The Reference discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

| No. | '740 Patent Claim 23 | The Reference |
|---|---|---|

| | | |
|---|---|---|
| 23 | The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. | The Reference discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

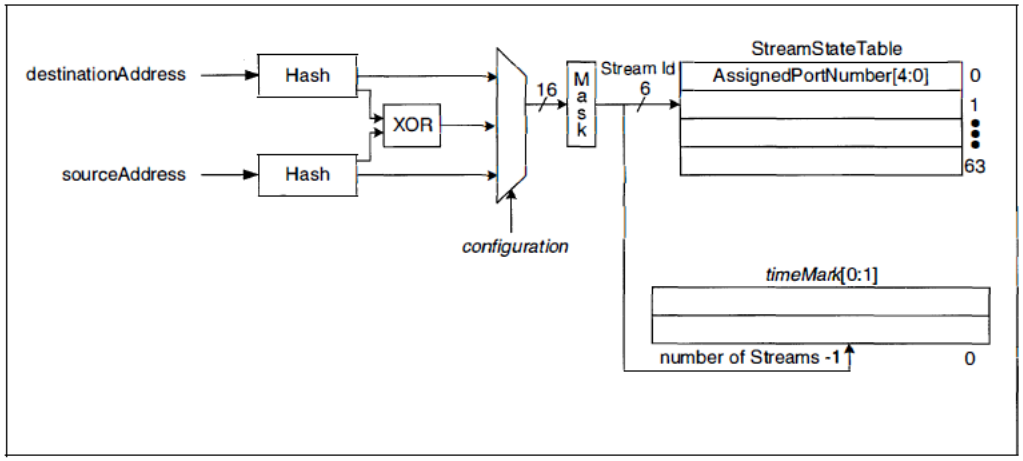



FIG. 2

DeJager '424 at Figure 3A

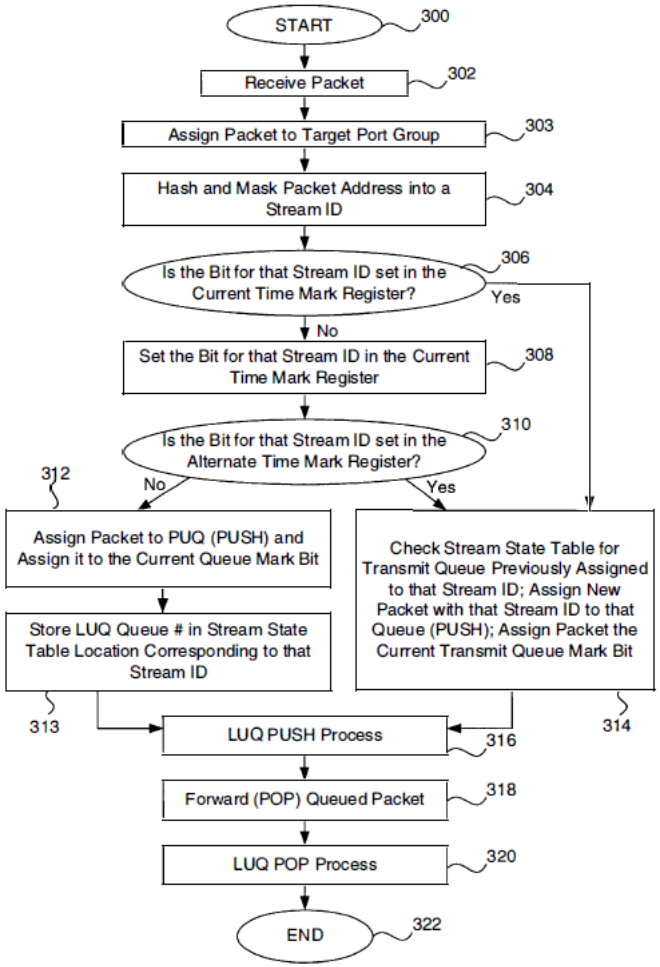



***FIG. 3A***

DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a

| | | |
|---|---|---|
| | | mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the |

| | | |
|---|---|---|
| | | negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. |

| | | |
|---|---|---|
| | | When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.") |

| | | |
|---|---|---|
| | | Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates |

| | | |
|---|---|---|
| | | information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

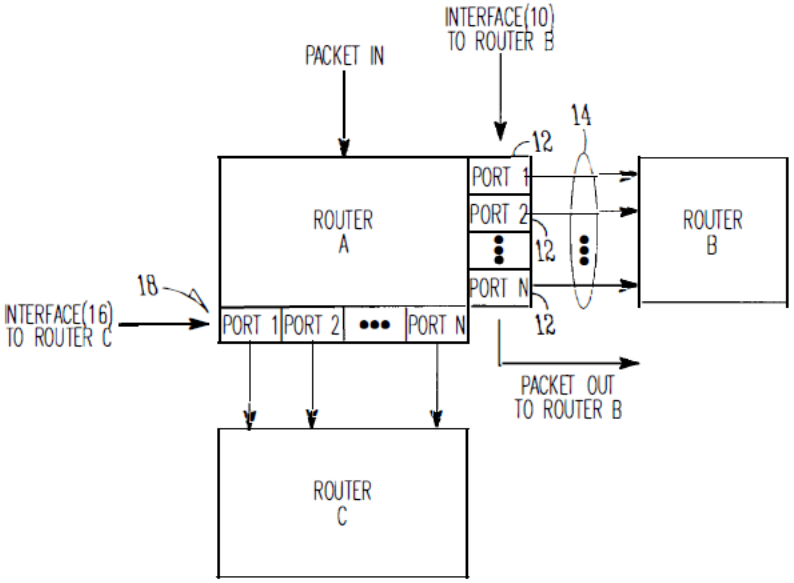
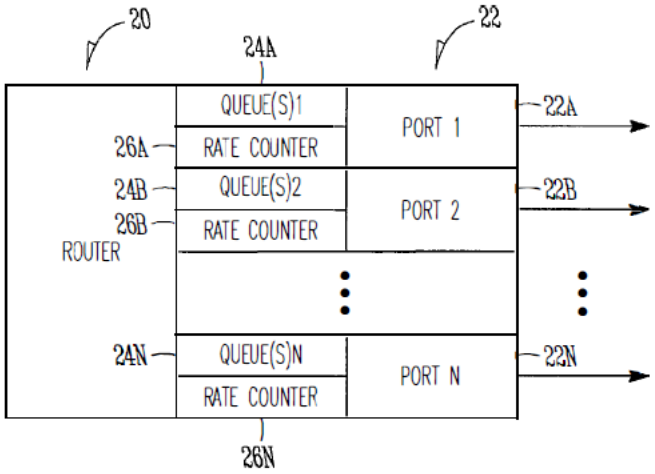
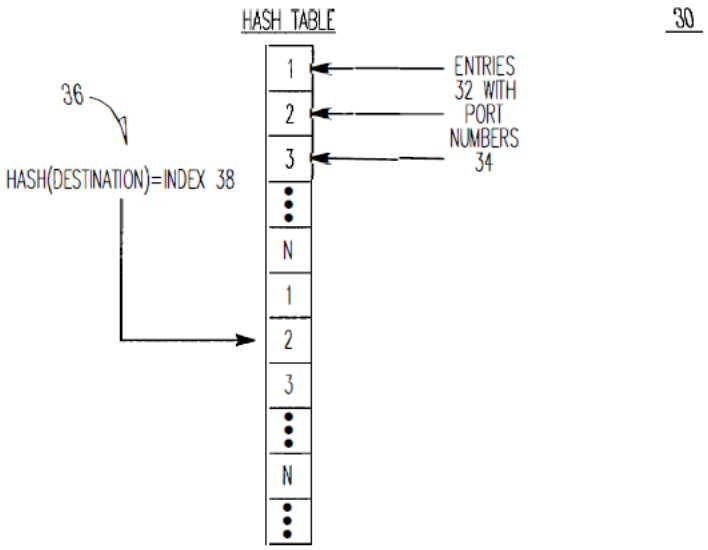
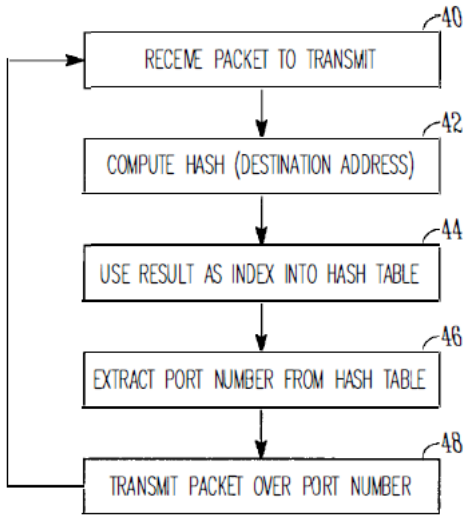


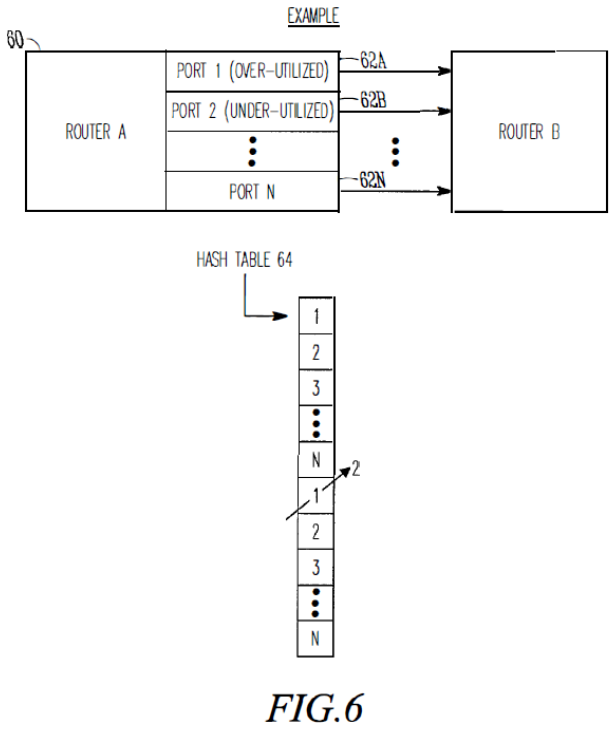

*FIG. 1*

Li '914 at Figure 2

| | | |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| | | |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| | | |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| | | |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

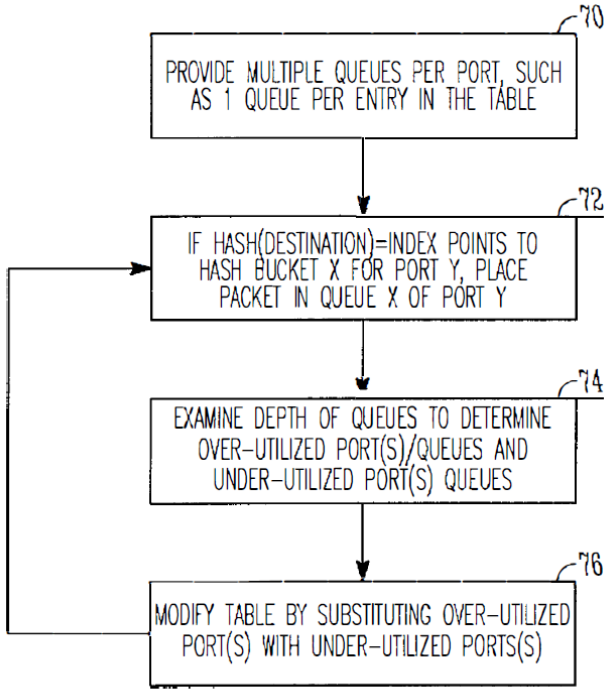



*FIG. 7*

Li '914 at Figure 8

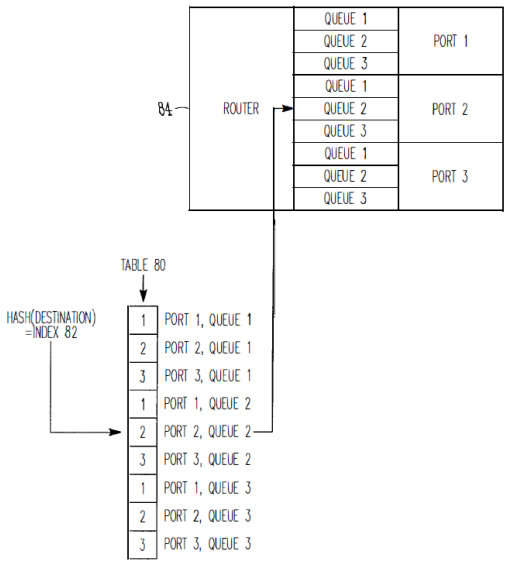



*FIG. 8*

Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")

Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the

| | | |
|---|---|---|
| | | above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash |

| | | |
|---|---|---|
| | | operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed |

| | | |
|---|---|---|
| | | multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link |

| | | |
|---|---|---|
| | | identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.") |

| | | |
|---|---|---|
| | | Borgione ’125 at 5:28-54 (“The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.”)<br><br>Borgione ’125 at 7:1-5 (“A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).”)<br><br>Borgione ’125 at Figure 2-5 |

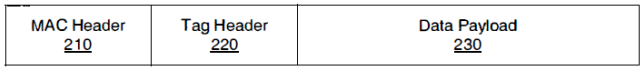



**Figure 2**




**Figure 3**

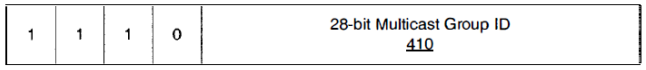



**Figure 4**

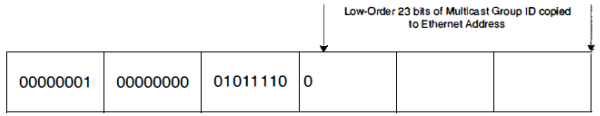
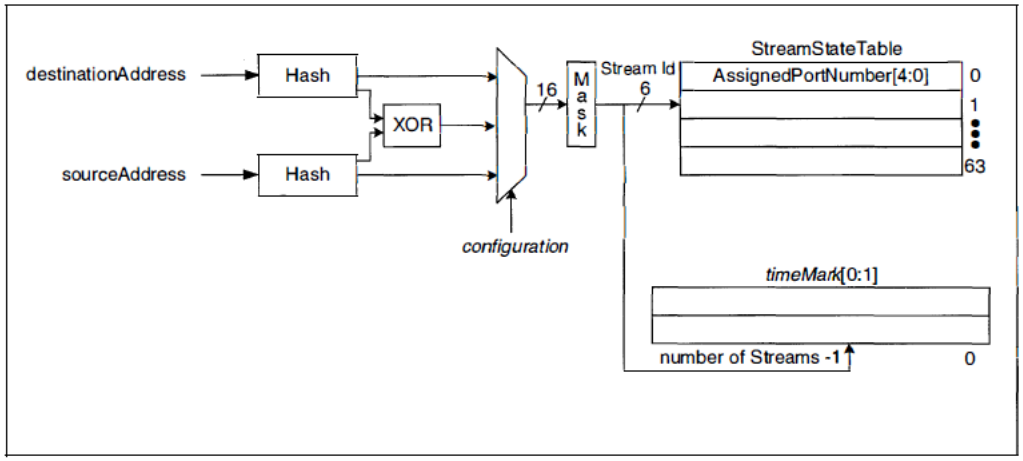



**Figure 5**

| No. | '740 Patent Claim 24 | The Reference |
|---|---|---|

| | | |
|---|---|---|
| 24 | The apparatus according to claim 23, wherein the mapping function comprises a hashing function. | The Reference discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| | | |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

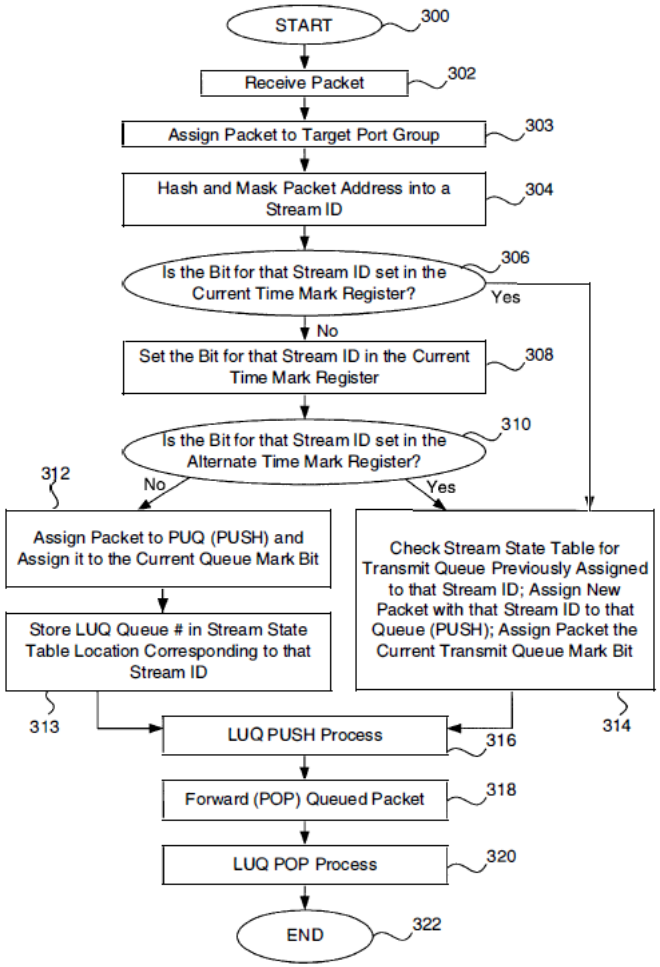



***FIG. 3A***

DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a

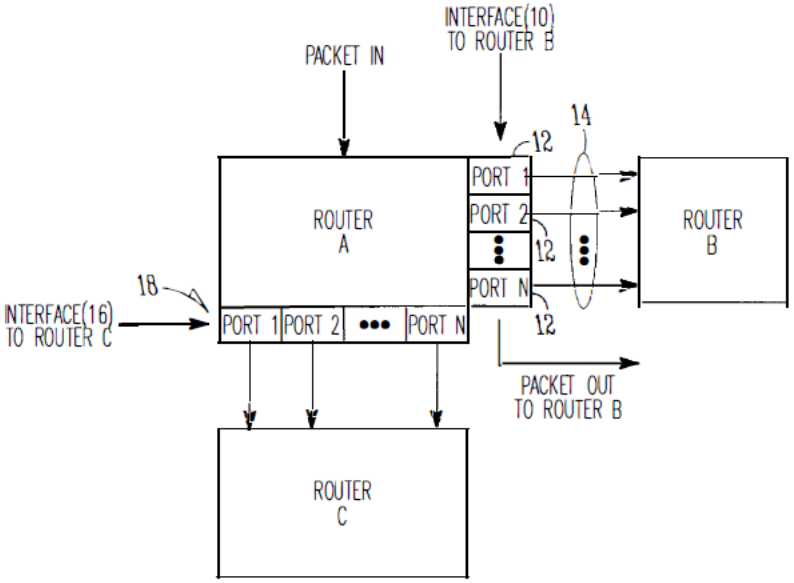
| | | |
|---|---|---|
| | | mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the |

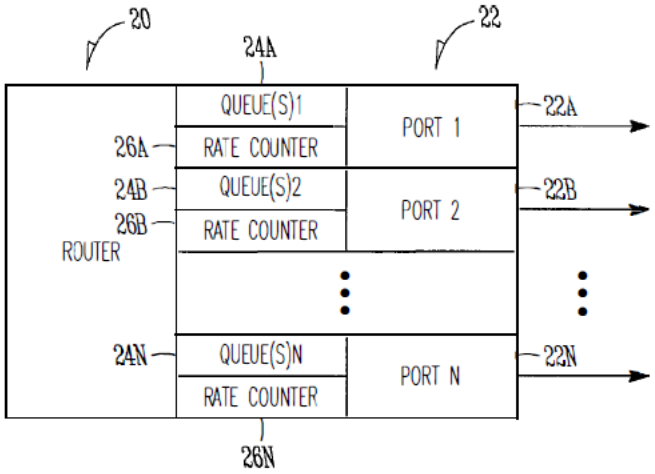
| | | |
|---|---|---|
| | | negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. |

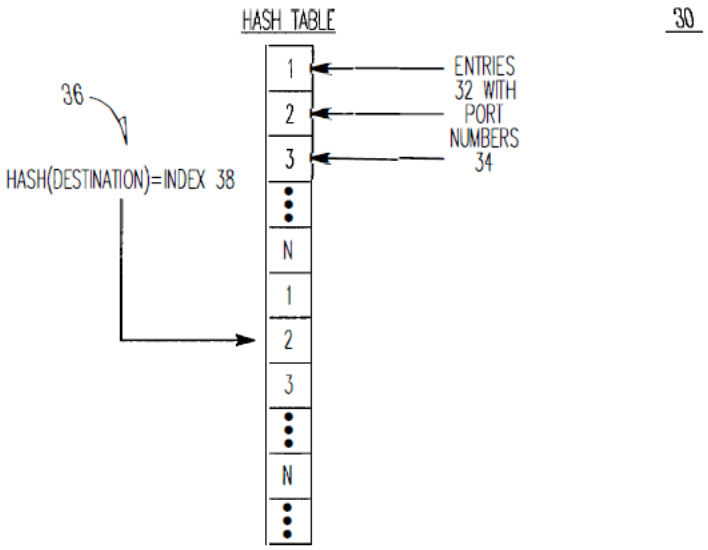
| | | |
|---|---|---|
| | | When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.") |

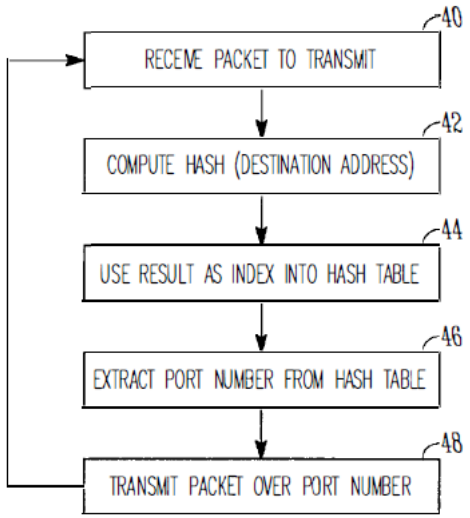
| | | |
|---|---|---|
| | | Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates |

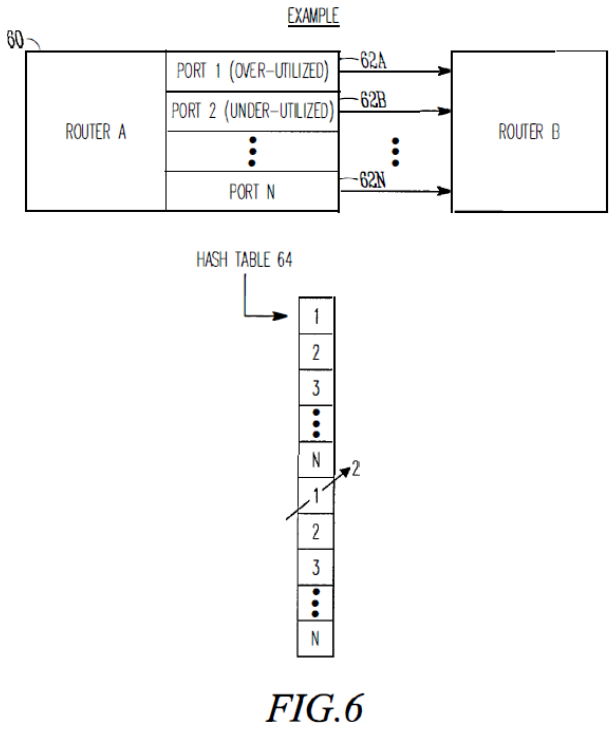
| | | |
|---|---|---|
| | | information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

| | | |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

| | | |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| | | |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| | | |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| | | |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

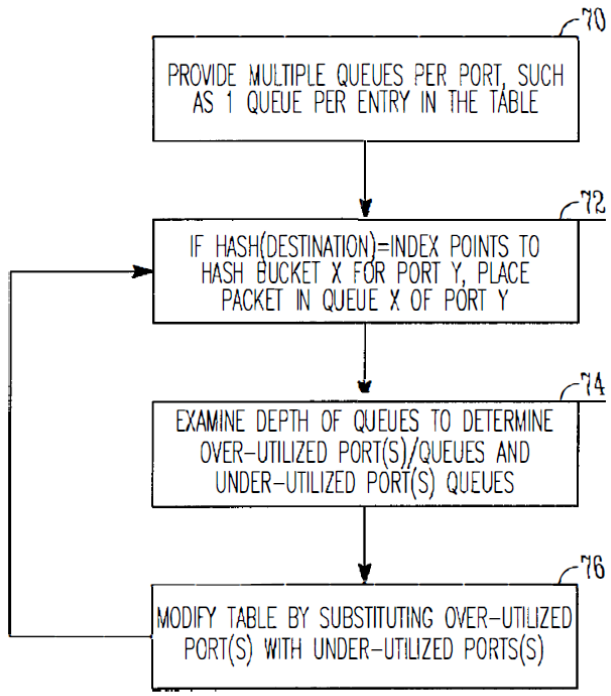



*FIG. 7*

Li '914 at Figure 8

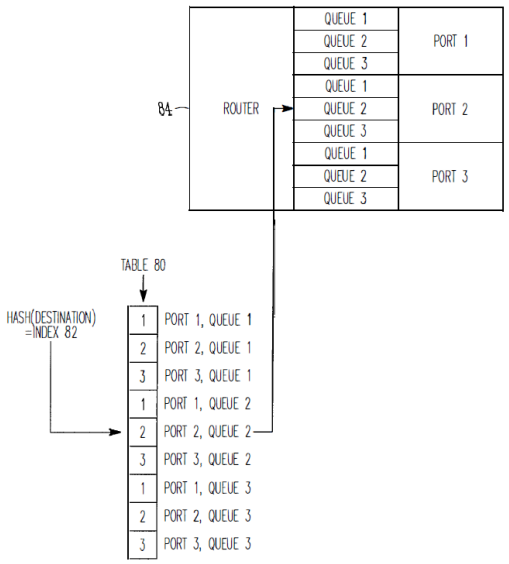



*FIG. 8*

Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")

Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the

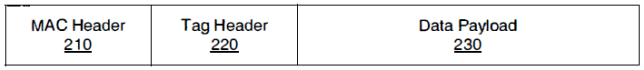
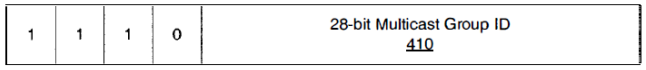
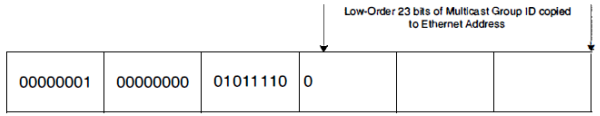
| | | |
|---|---|---|
| | | above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing.<br>In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash |

| | | |
|---|---|---|
| | | operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed |

| | | |
|---|---|---|
| | | multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link |

| | | |
|---|---|---|
| | | identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.") |

| | | |
|---|---|---|
| | | Borgione ’125 at 5:28-54 (“The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.”)<br><br>Borgione ’125 at 7:1-5 (“A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).”)<br><br>Borgione ’125 at Figure 2-5 |

| | | |
|---|---|---|
| | |  <br>**Figure 2**<br>  <br>**Figure 3**<br>  <br>**Figure 4**<br>  <br>**Figure 5** |

| No. | '740 Patent Claim 25 | The Reference |
|---|---|---|
| 25[a] | The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a | The Reference discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

| No. | '740 Patent Claim 25 | The Reference |
|---|---|---|
| | number of at least some of the first and second physical links, | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 25[b] | to apply the hashing function to the at least one of the frame attributes to produce a hashing key, | The Reference discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 25[c] | to calculate a modulo of a division operation of the hashing key by the hashing size, and | The Reference discloses to calculate a modulo of a division operation of the hashing key by the hashing size.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, and Singh. |

| No. | '740 Patent Claim 25 | The Reference |
|---|---|---|
| | | Singh at 9:30-43 ("The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.")<br><br>Singh at 11:29-38 ("FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 16:28-44 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.") |

| No. | '740 Patent Claim 25 | The Reference |
|---|---|---|
| | | Singh at 17:31-49 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.") |
| 25[d] | to select the first and second physical links responsively to the modulo. | The Reference discloses to select the first and second physical links responsively to the modulo.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, and Singh.<br><br>Singh at 9:30-43 ("The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 |

| No. | '740 Patent Claim 25 | The Reference |
|---|---|---|
| | | ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.")<br><br>Singh at 11:29-38 ("FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 16:28-44 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.")<br><br>Singh at 17:31-49 ("In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on |

| No. | '740 Patent Claim 25 | The Reference |
|---|---|---|
| | | the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.") |

| No. | '740 Patent Claim 26 | The Reference |
|---|---|---|
| 26 | The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. | The Reference discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, and Solomon.<br><br>Solomon at [0054] ("Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.")<br><br>Solomon at [0056] ("Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which |

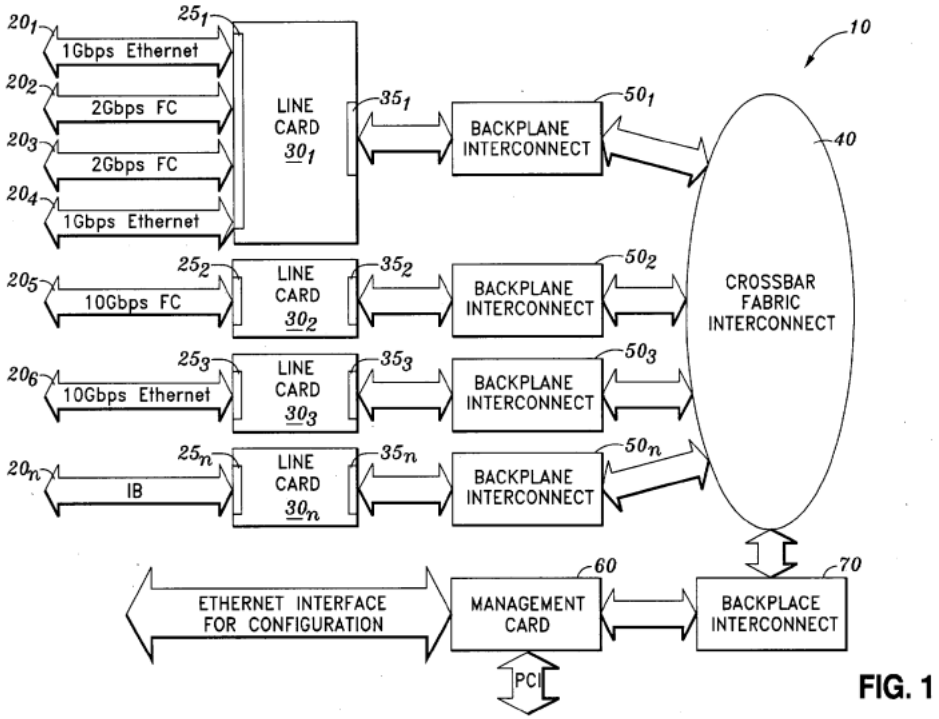
| | | |
|---|---|---|
| | | was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.") |

| No. | '740 Patent Claim 27 | The Reference |
|---|---|---|
| 27 | The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. | The Reference discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

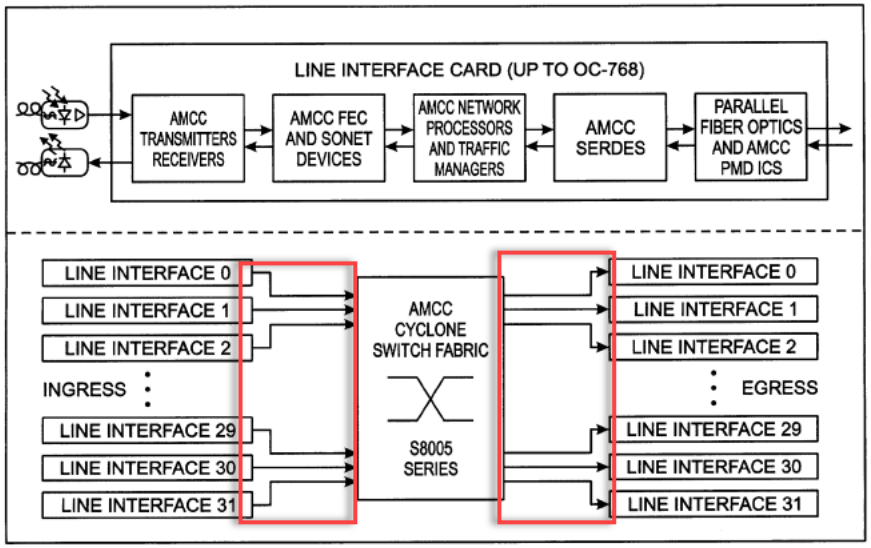
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| 28[preamble] | Apparatus for connecting a network | The Reference discloses apparatus for connecting a network node with a communication network. |

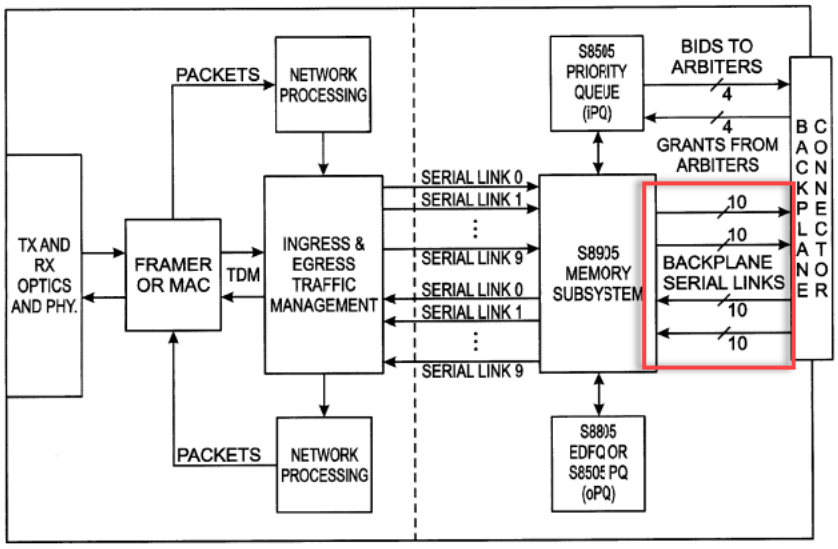
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | node with a communication network, comprising: | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 28[a] | one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network; | The Reference discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 28[b] | a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules; | The Reference discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel |

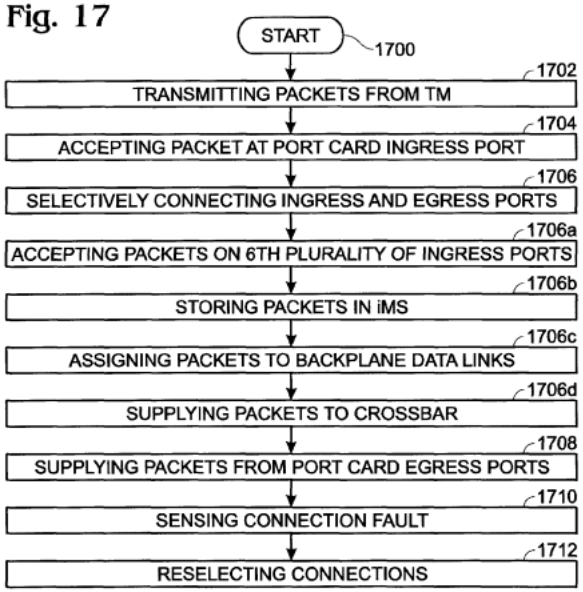
| No. | ’740 Patent Claim 28 | The Reference |
|---|---|---|
| | | System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher ’530. |
| 28[c] | a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and | The Reference discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher ’530, Viswanathan, Singh, Smith ’430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] (“In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.”) |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1<br>  <br><br>FIG. 1 |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

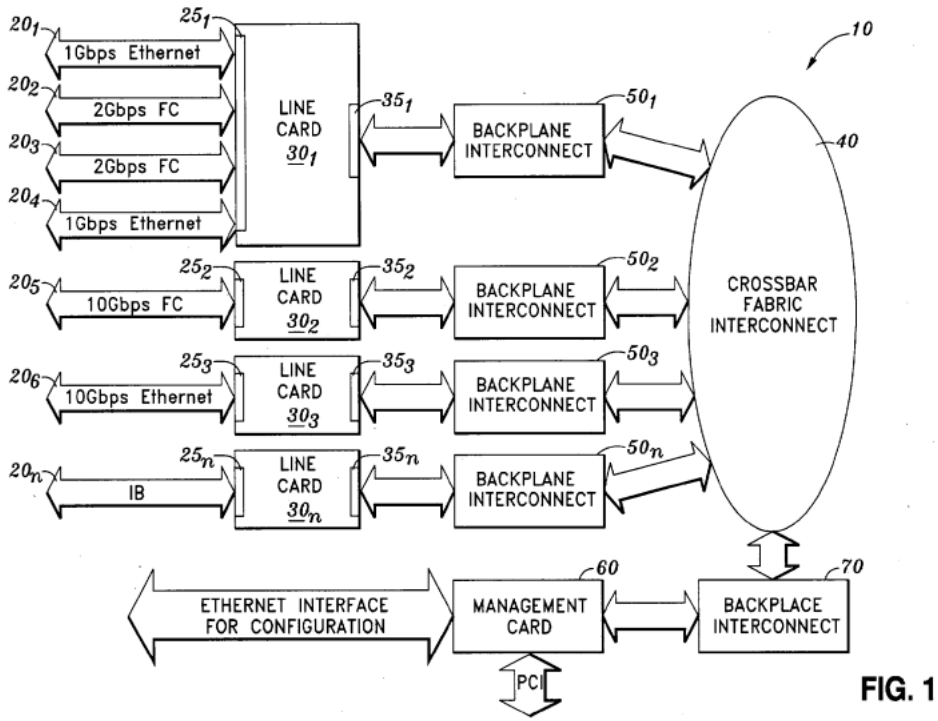
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

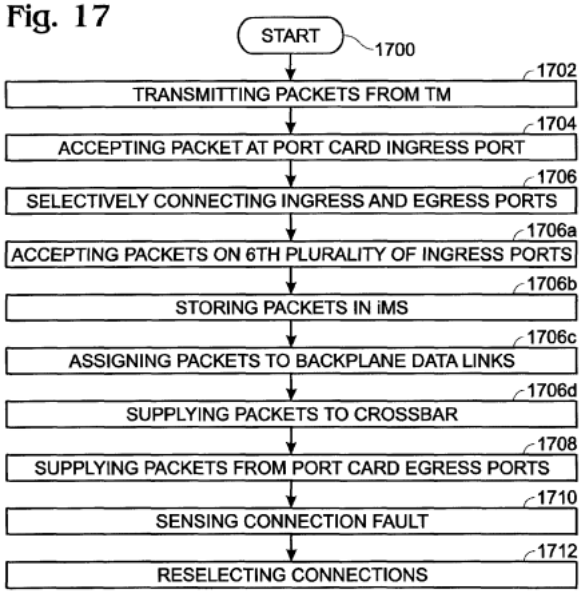
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 28[d] | a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame, | The Reference discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces |

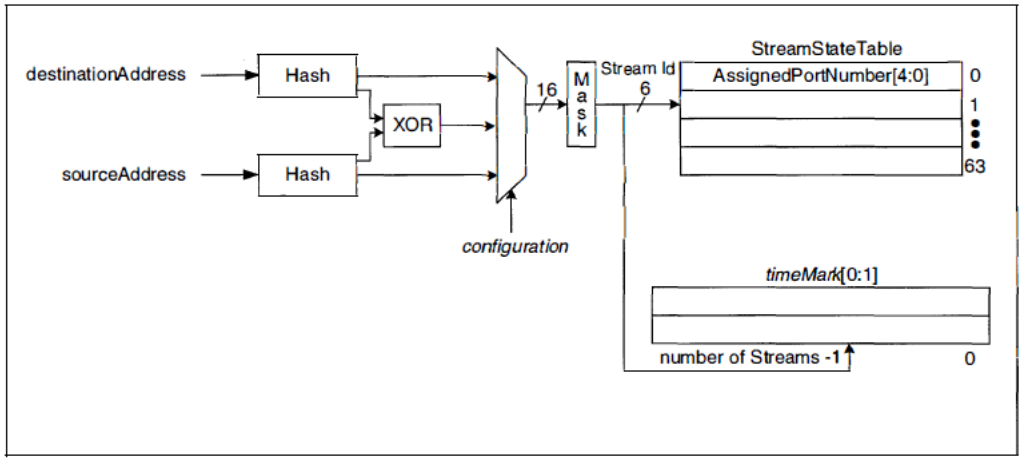
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

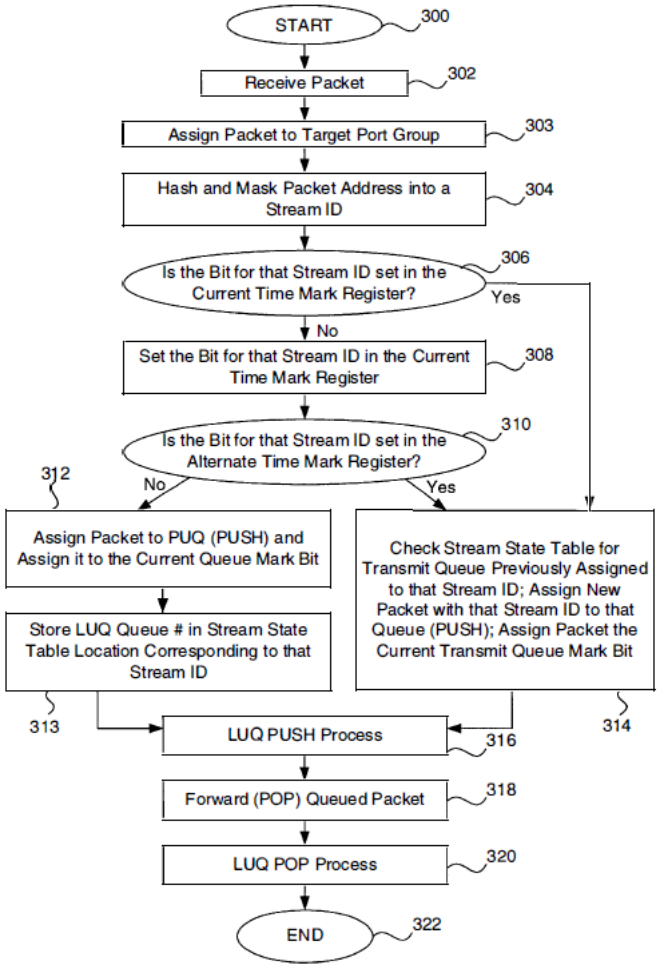
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | • Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

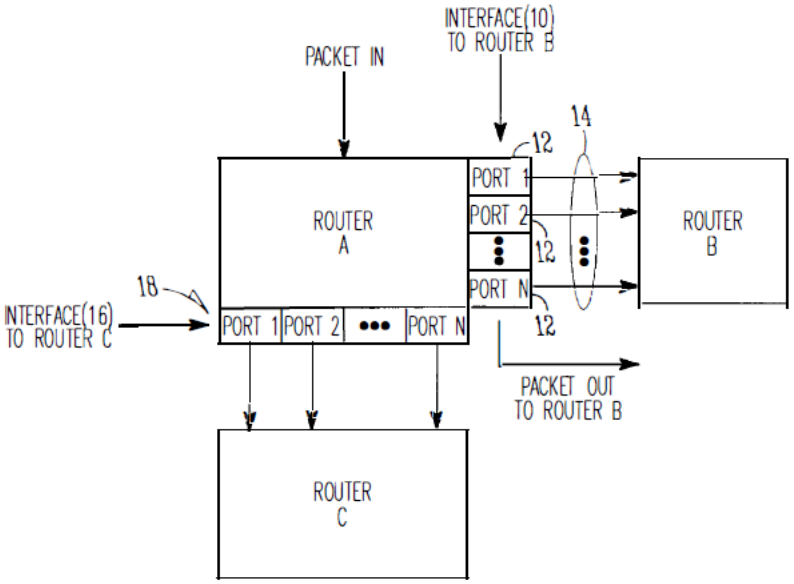
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

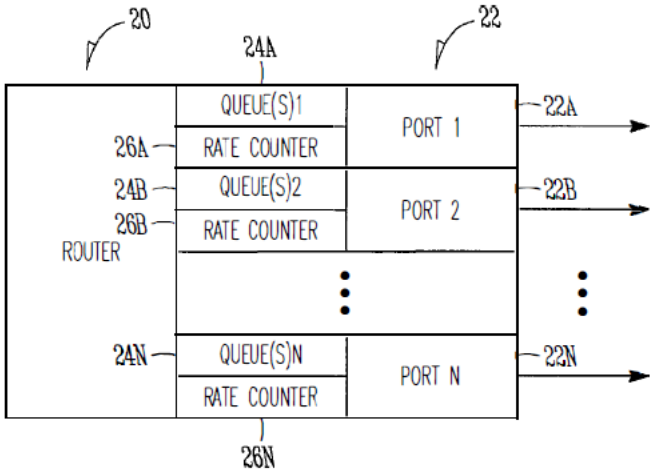
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

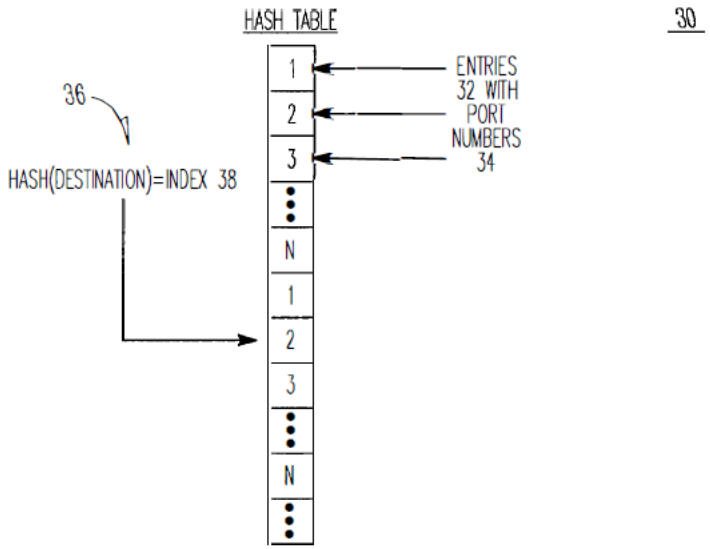
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

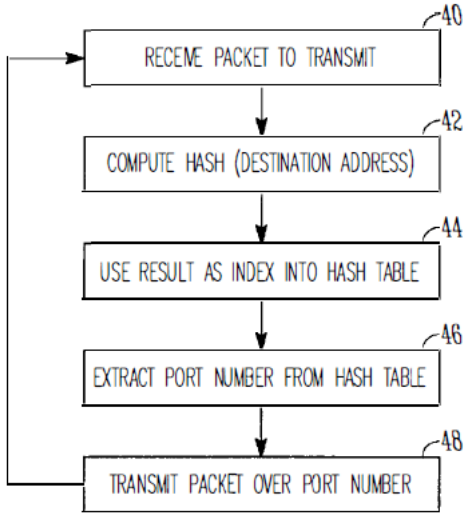
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

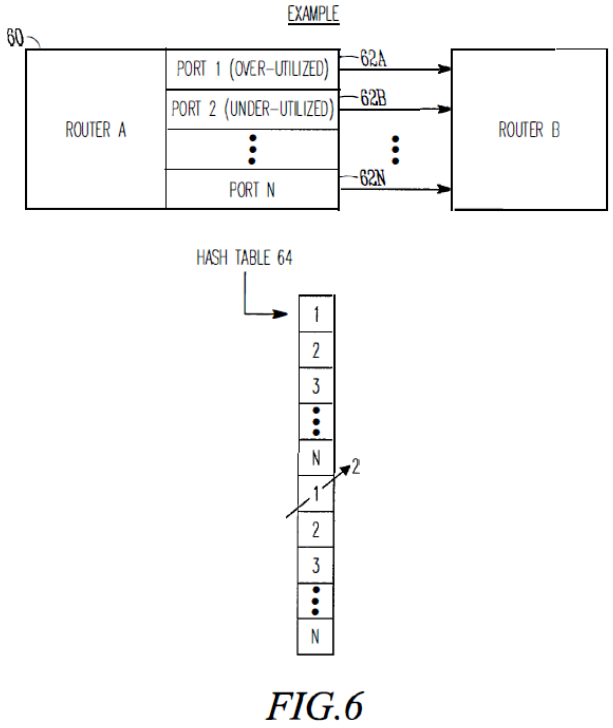
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

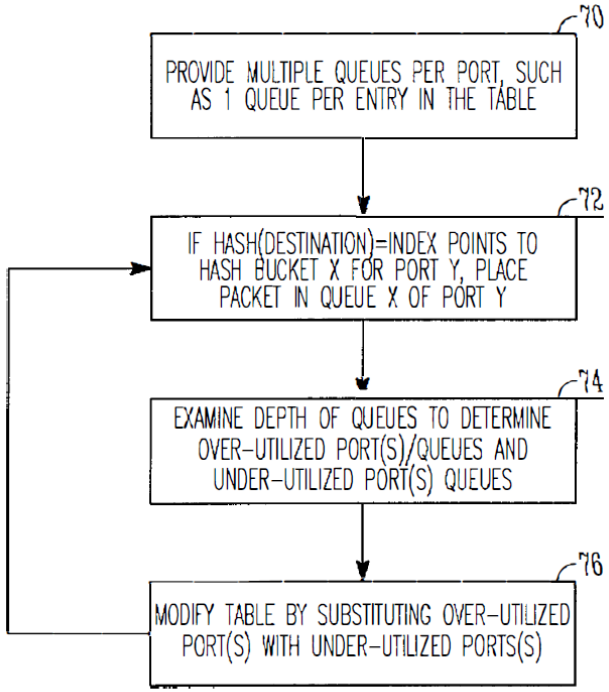
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

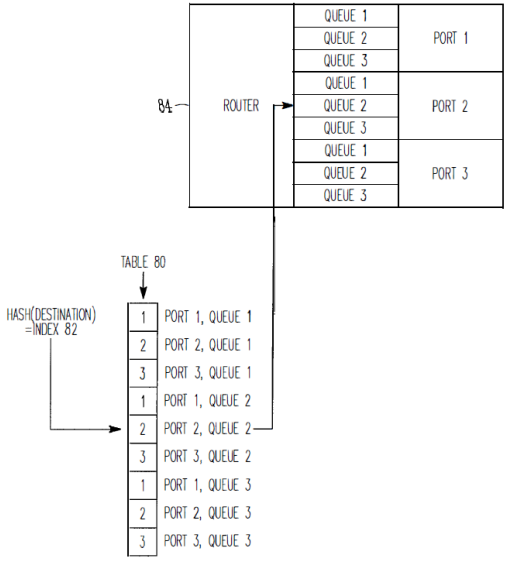
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

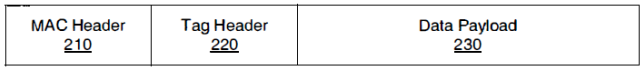
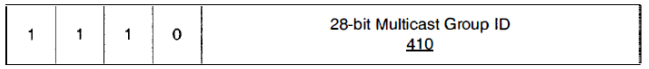
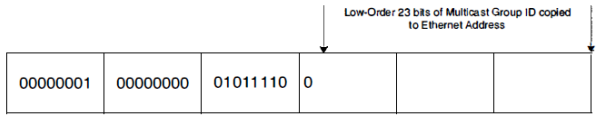
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |
| 28[e] | the communication network being | The Reference discloses the communication network being arranged to provide a communication service to the network node. |

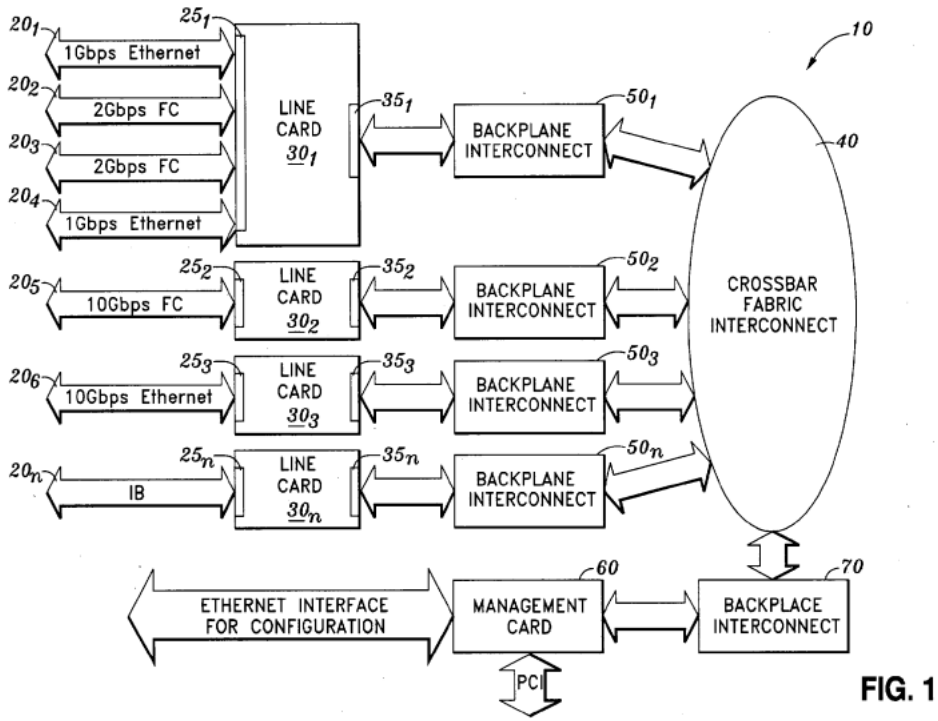
| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | arranged to provide a communication service to the network node, | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 28[f] | the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and | The Reference discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR).<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 28[g] | the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively | The Reference discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the band width requirements.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

| No. | '740 Patent Claim 28 | The Reference |
|---|---|---|
| | to the band width requirements. | skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

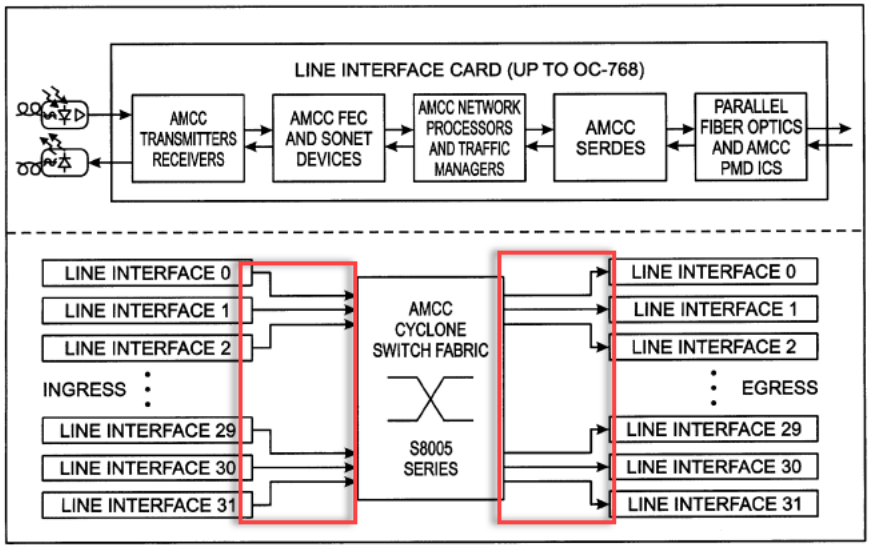
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| 29[preamble] | Apparatus for connecting user ports to a communication network, comprising: | The Reference discloses apparatus for connecting user ports to a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 29[a] | one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network, | The Reference discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |

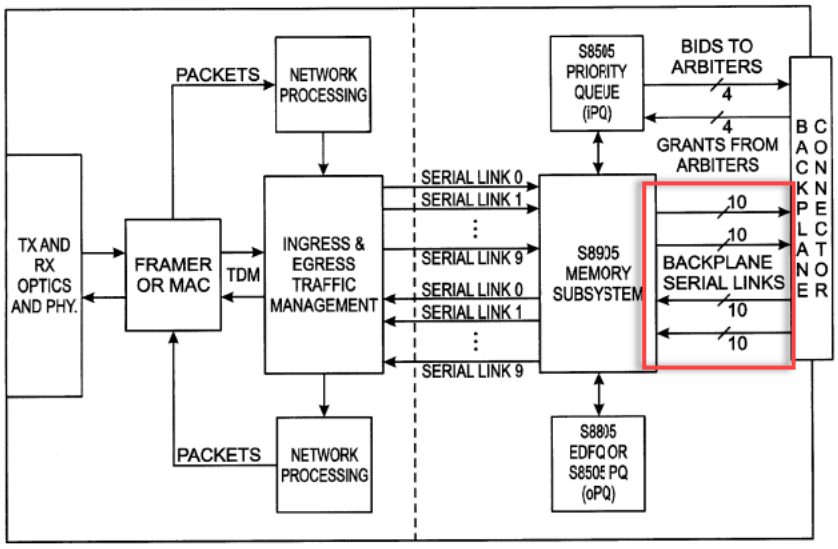
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | |
| 29[b] | at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction; | The Reference discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 29[c] | a backplane having the one or more user interface comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network, | The Reference discloses a backplane having the one or more user interface comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces |

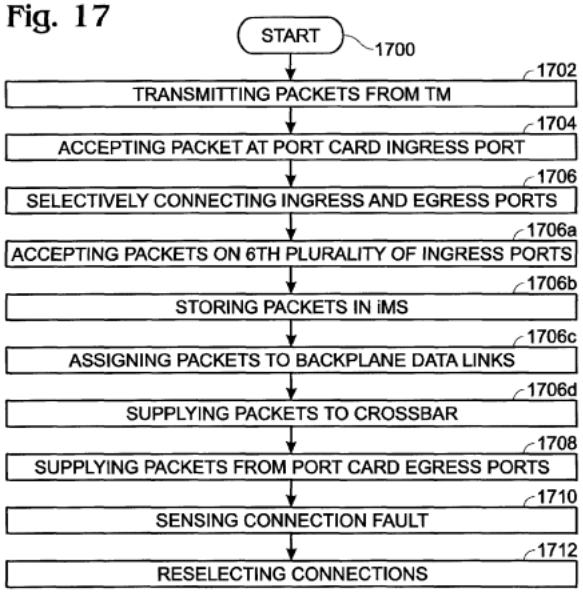
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

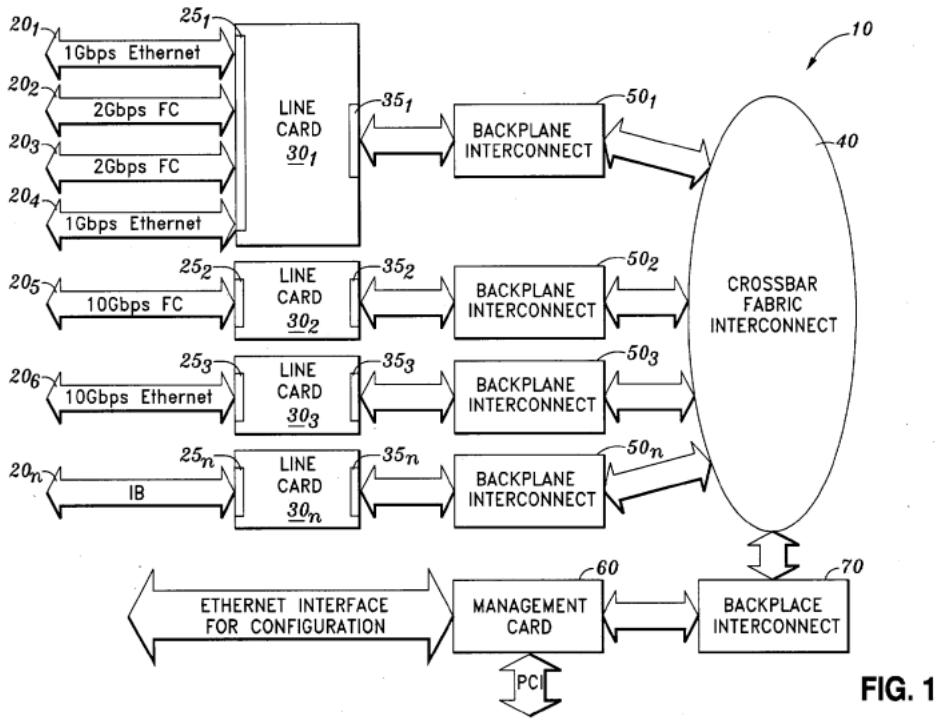
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

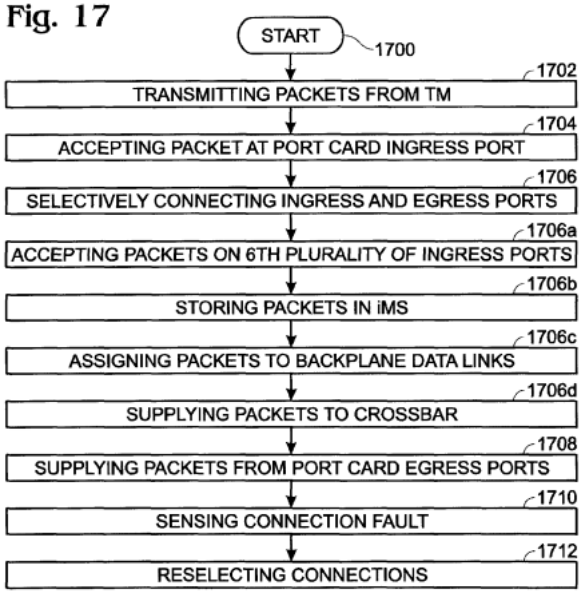
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 29[d] | at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and | The Reference discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 29[e] | a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of | The Reference discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

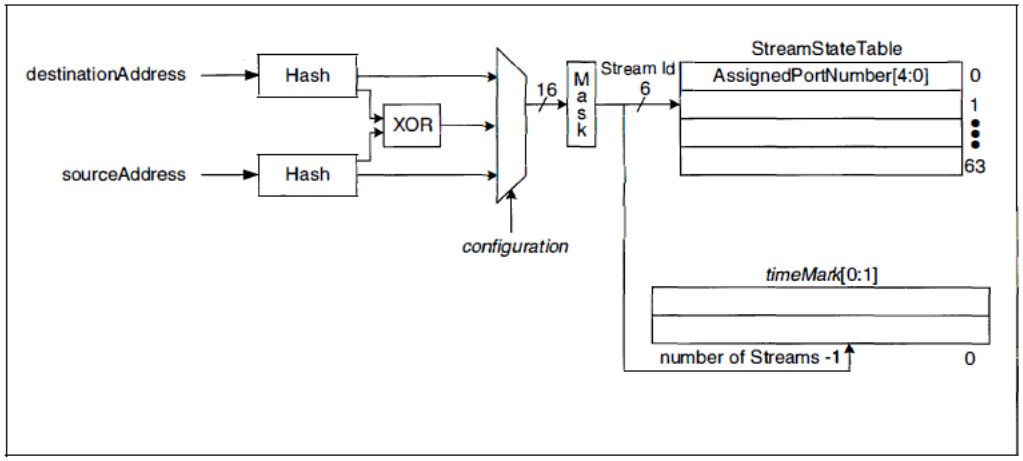
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | backplane traces over which to send the data frame. | skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

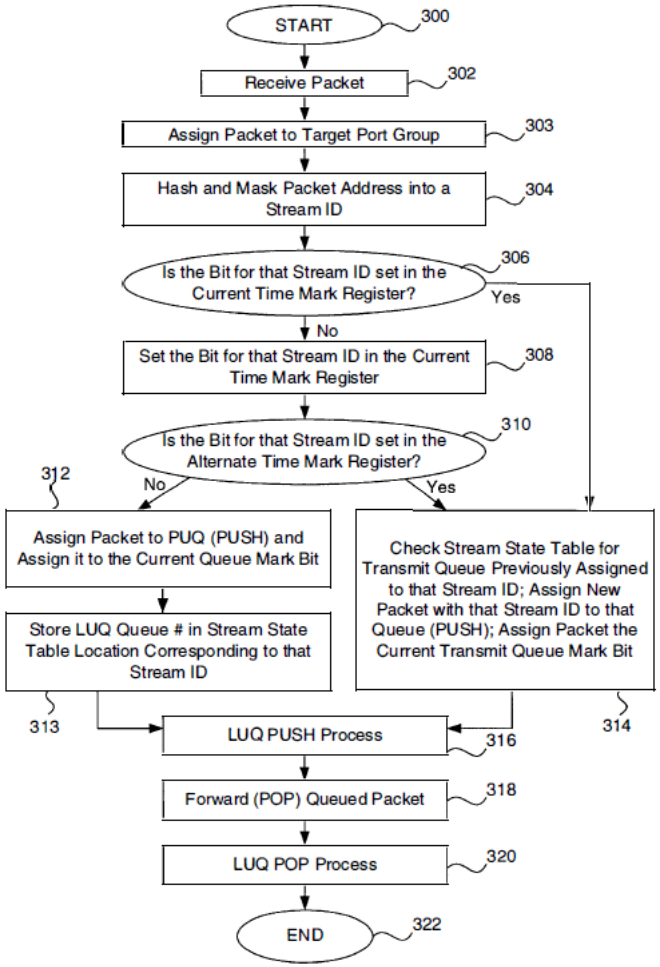
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | • Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br><br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

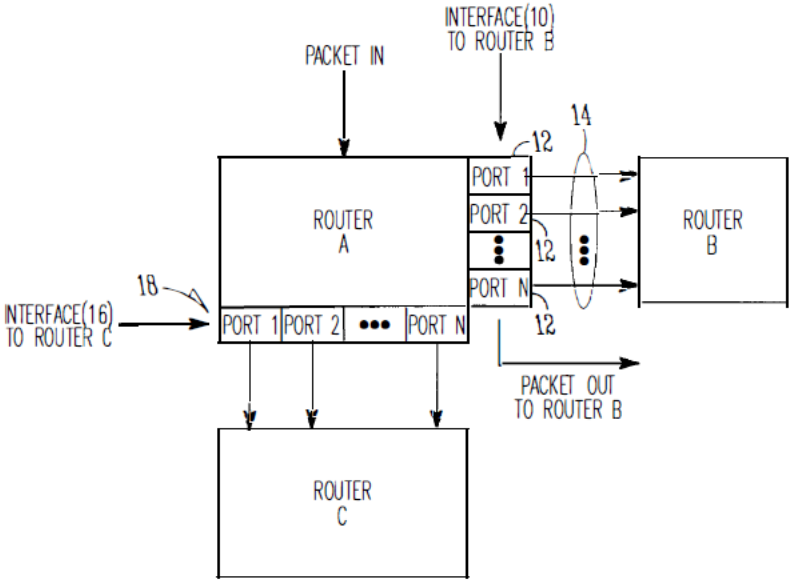
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

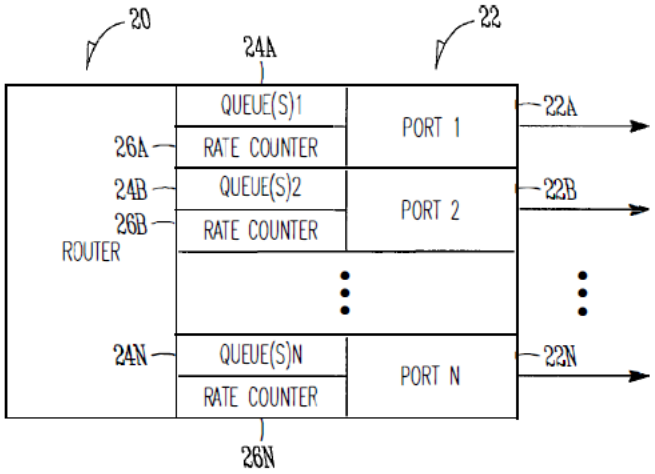
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

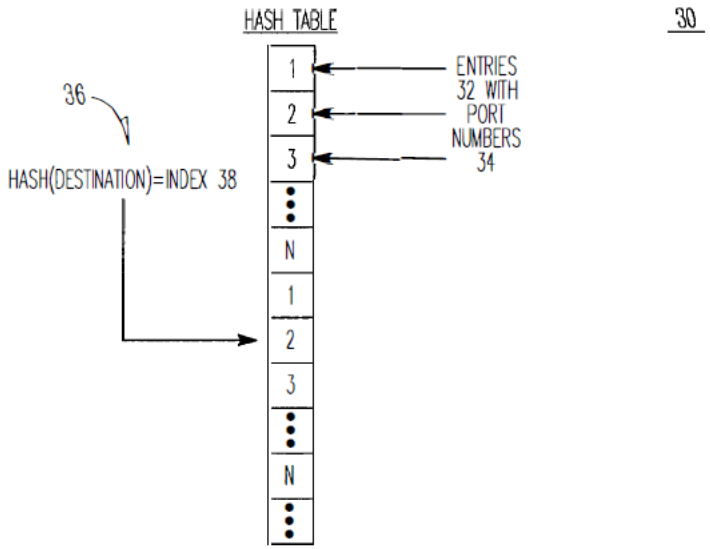
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

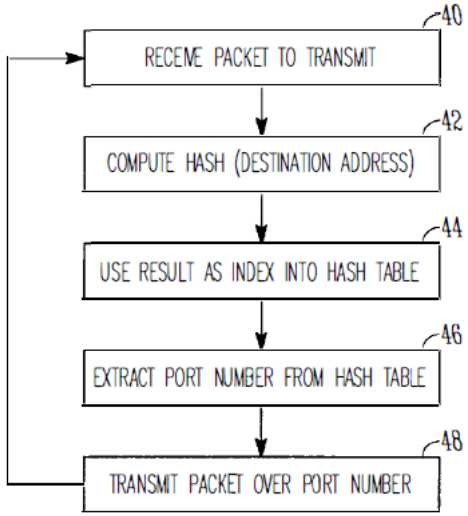
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

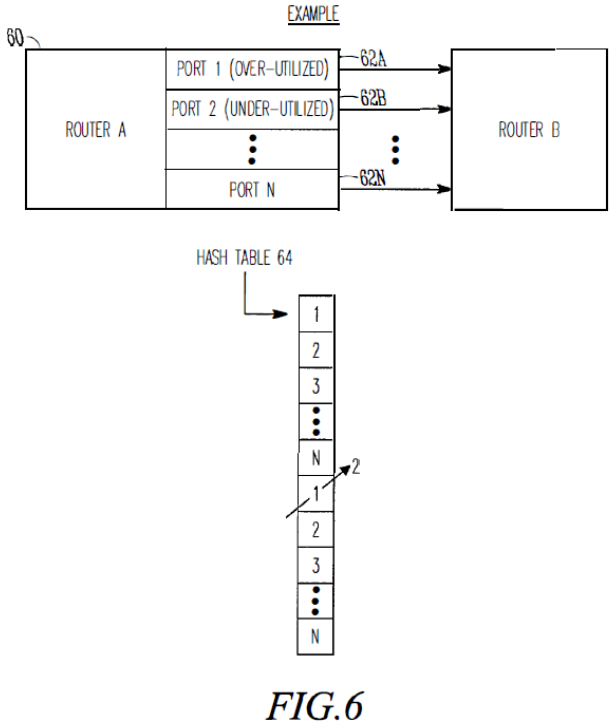
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

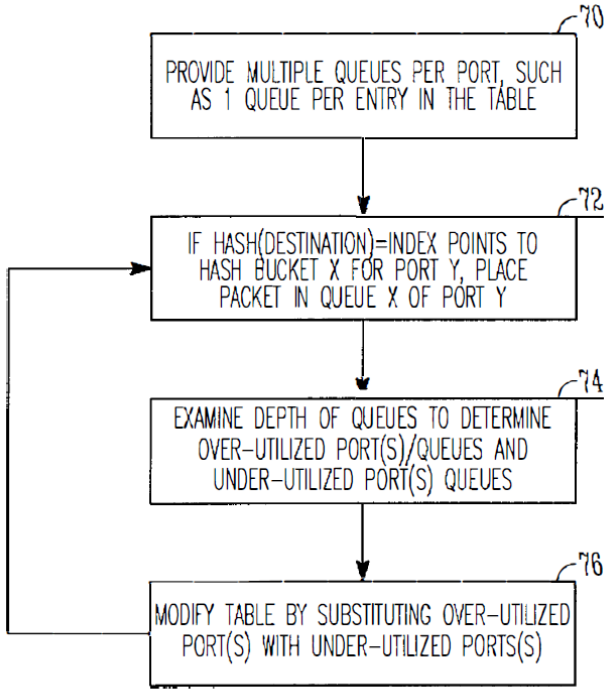
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

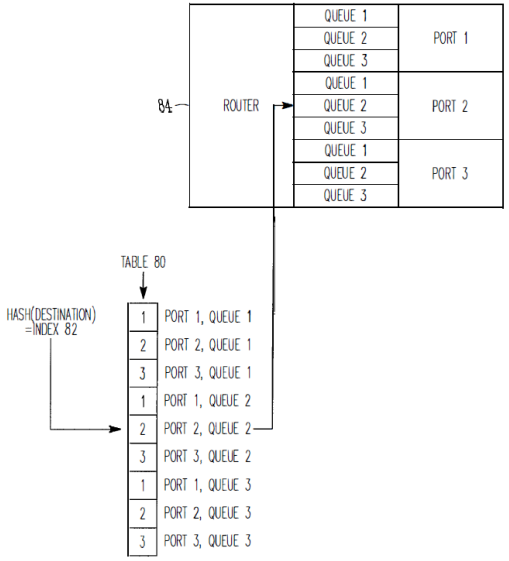
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

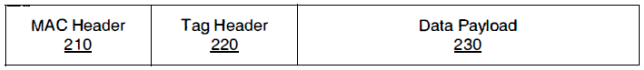
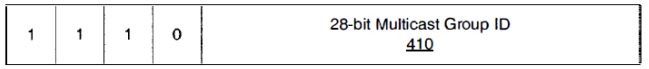
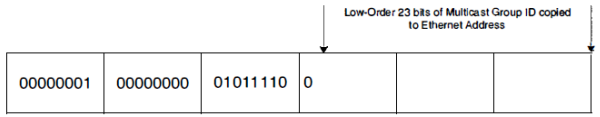
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

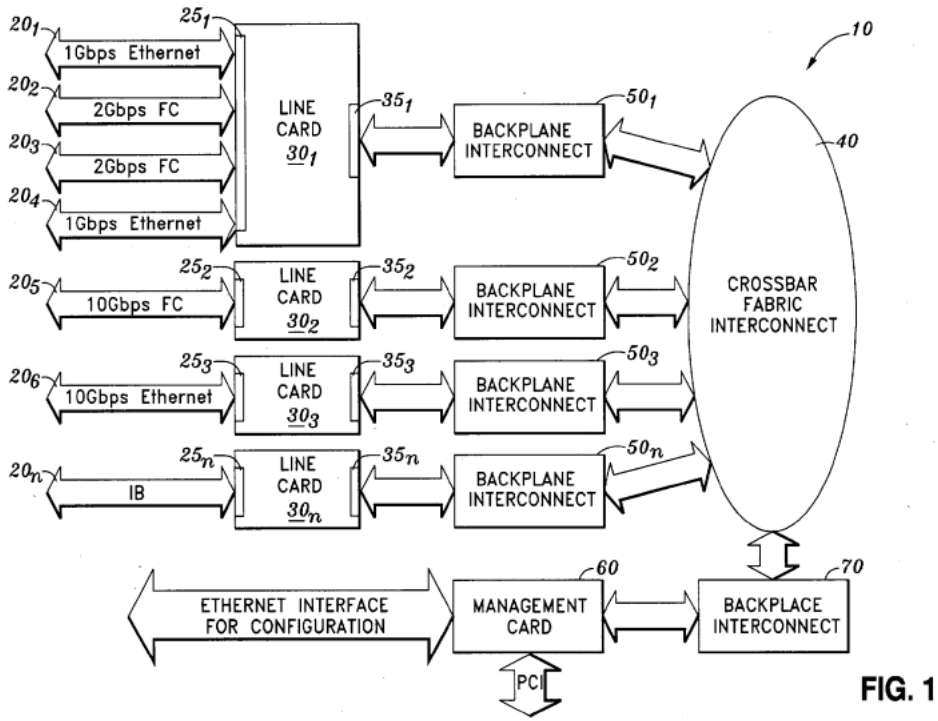
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

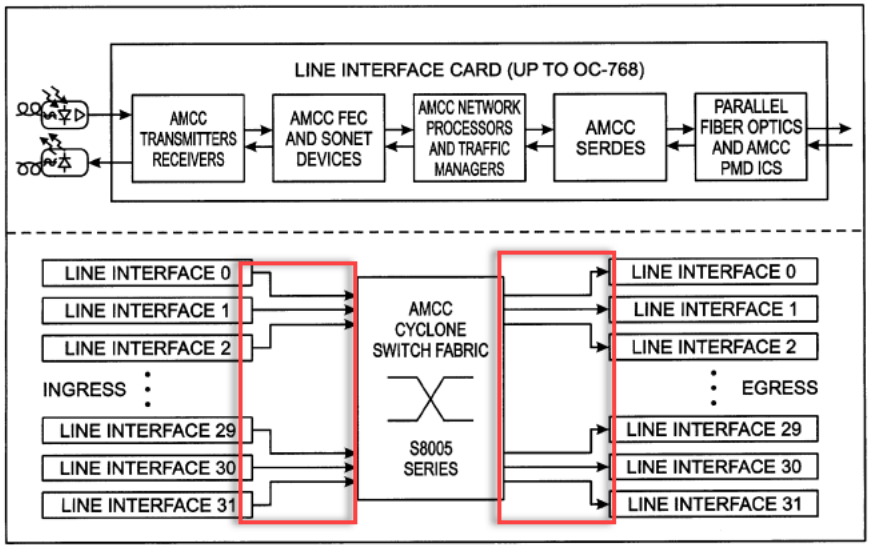
| No. | '740 Patent Claim 29 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

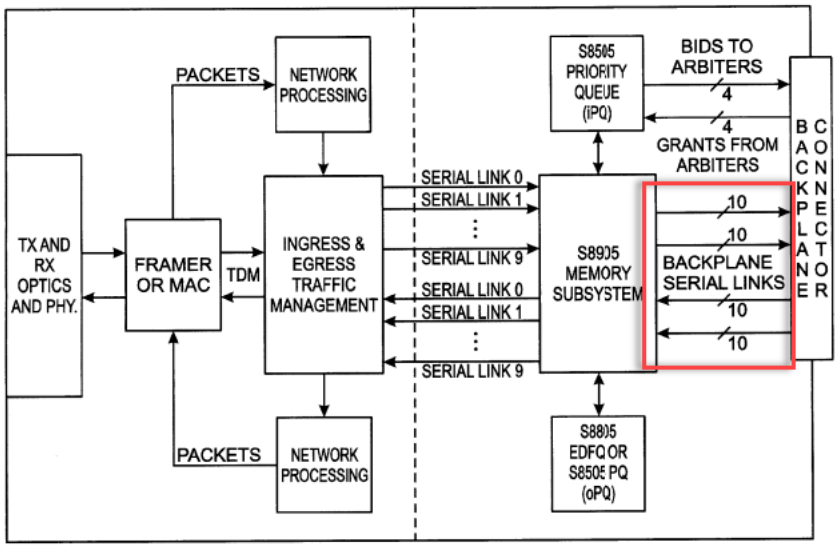
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| 30[preamble] | Apparatus for connecting user ports to a communication network, comprising: | The Reference discloses apparatus for connecting user ports to a communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 30[a] | one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network; | The Reference discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530. |
| 30[b] | a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to | The Reference discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

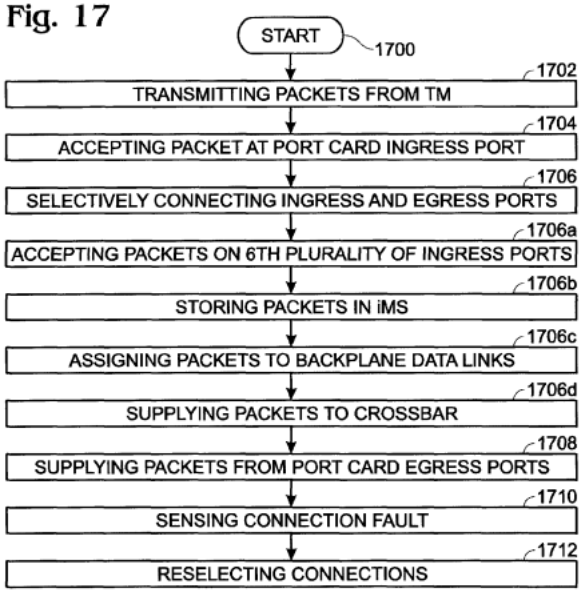
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | transfer the data frames between the one or more user interface modules and the communication network; | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, and Dontu.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Func-tional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

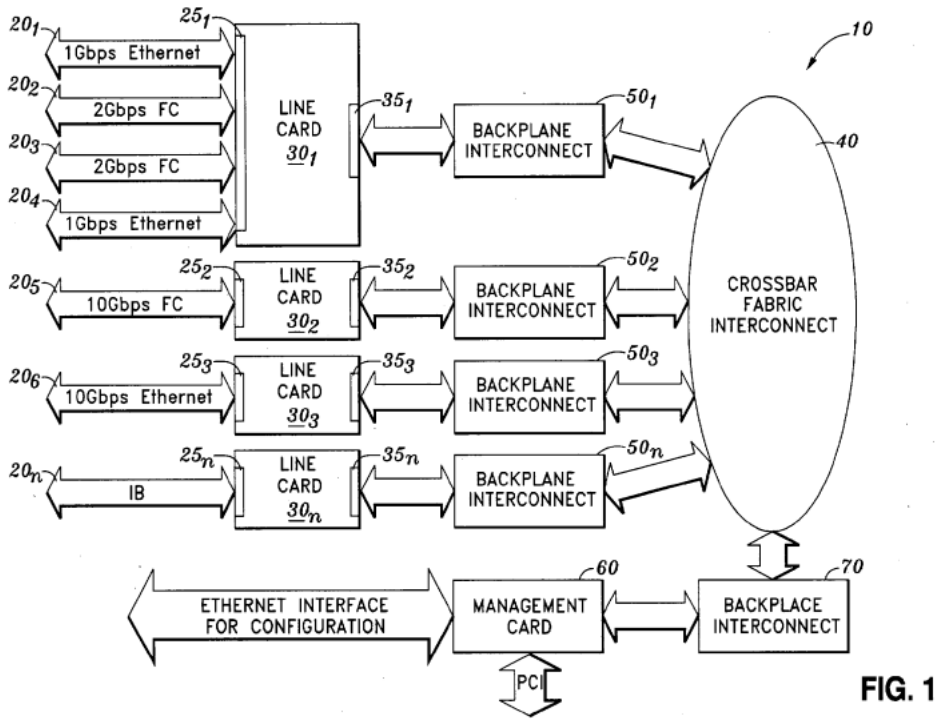
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

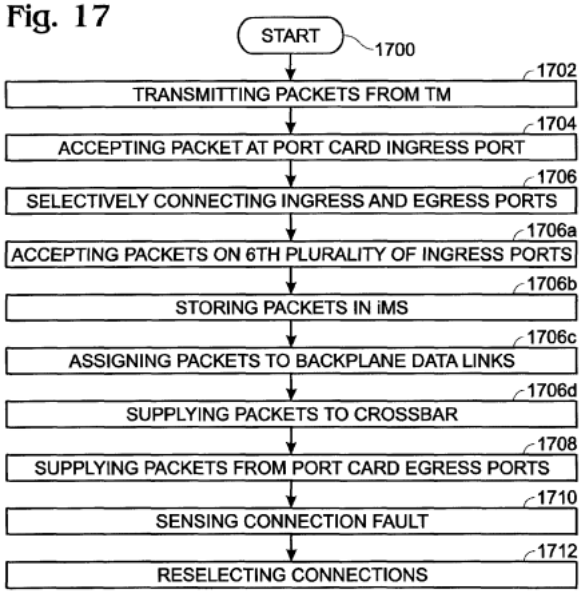
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.") |
| 30[c] | a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame; | The Reference discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be imple-mented as one or more line cards in a networked environ-ment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collec-tively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards |

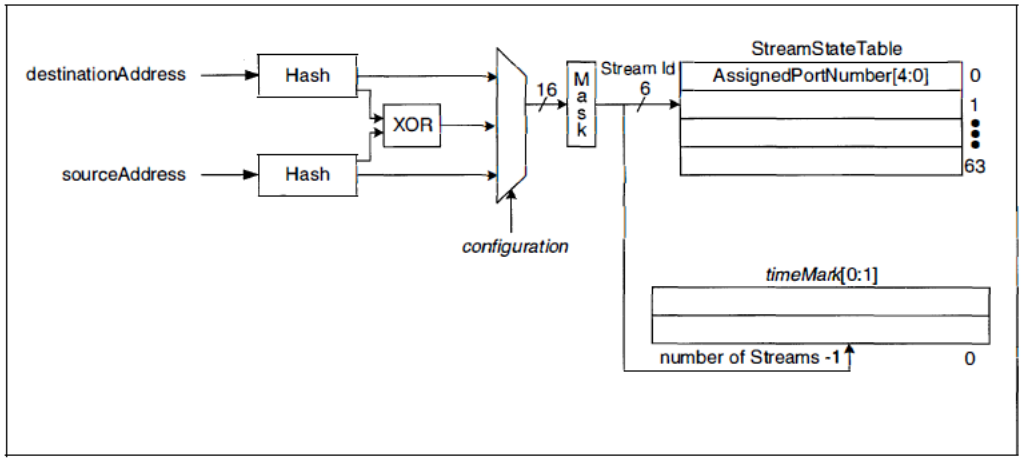
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar inter-connect 40 via backplane interconnects 501-50n (collec-tively, "backplane interconnects 30"). It should be appreci-ated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communi-cation across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus archi-tecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

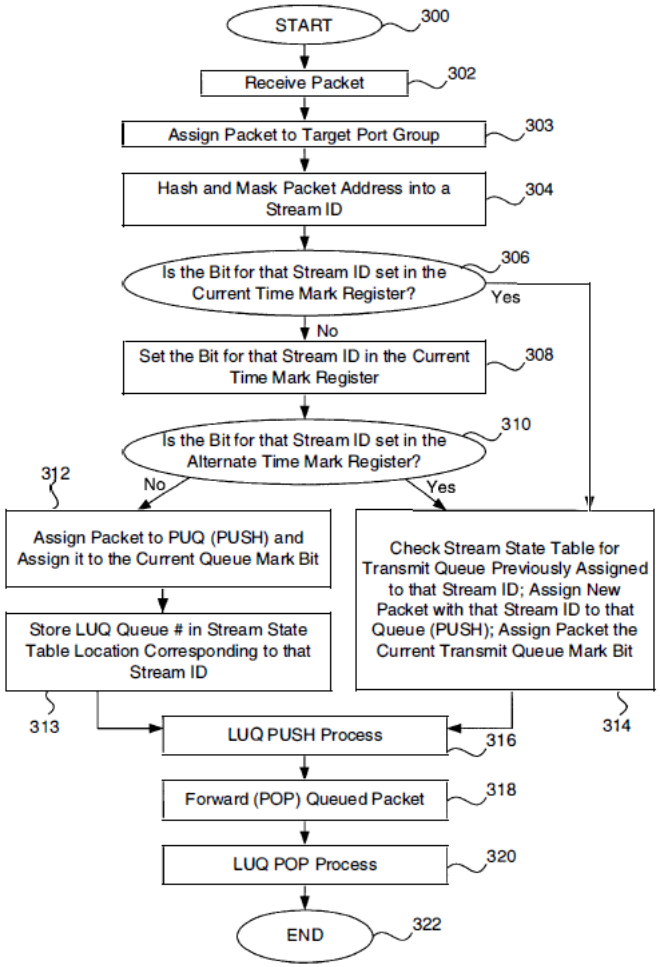
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress<br>data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.") |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Singh at 11:28-38 ("FIG. 9 is a diagram illustrating link to channel assign- ments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a charmel, whether it is a backplane channel or an ingress/egress charmel. There is no logical relationship 35 between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.")<br><br>Singh at 13:35-48 ("FIG. 10 is a diagram depicting iPQ arbiter interface to switchplane and backplane chamiel mapping. The arbiter interfaces on the iPQ directly correspond to the backplane channels of the MS, as shown. In other words, arbiter interfaces 0.A and 0.B handles the bids and grants for backplane chamiel 0. The two arbiters attached to interfaces 0.A and 0.B form switchplane O (as shown in FIG. 20) that controls the crossbars attached to the links of backplane channel 0. An iPQ has 8 arbiter interfaces and can handle the bids and grants to 4 switch planes, thus servicing all the 4 backplane channels possible in an MS. A 4-ingress channel configuration, shown in FIG. 6, requires two iPQs and two MSs to support a 2x speedup (generates 8 backplane chan-nels).")<br><br>Singh at 18:44-53 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes substeps. Step 1706a includes each port card accepting packets on a sixth plurality of ingress data links, through a corresponding sixth plurality of port card ingress ports, from at least one ingress TM (iTM). Step 1706b stores the accepted packets in a port card ingress memory sub- system (iMS). Step 1706c assigns packets to a second plurality of port card backplane data links. Step 1706d supplies assigned packets to a crossbar.")<br><br>Singh at 18:61-19:9 ("The egress function of the switch fabric works analo-gously to the ingress function. Although the substeps asso-ciated with the egress function are listed below, they are not included in the figure in the interest of clarity. In some aspects, selectively connecting port card ingress ports to port card egress ports in Step 1706 includes additional substeps. Step 1706e includes each port card accepting packets on a second plurality of port |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | card backplane data links from crossbars. Step 1706/ stores the accepted packets in a port card egress memory subsystem ( eMS). Step 1706g assigns packets to a sixth plurality of port card egress ports. Step 1706h supplies assigned packets to selected port card egress 5 ports from the eMS. Step 1706i includes each port card supplying packets on a sixth plurality of egress data links, through the corresponding sixth plurality of port card ports, to at least one egress TM (eTM).")<br><br>Singh at Figure 17<br><br>Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | handling a packet addressed to server 104(3), net-work device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | • Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2<br><br>  <br><br>FIG. 2<br><br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

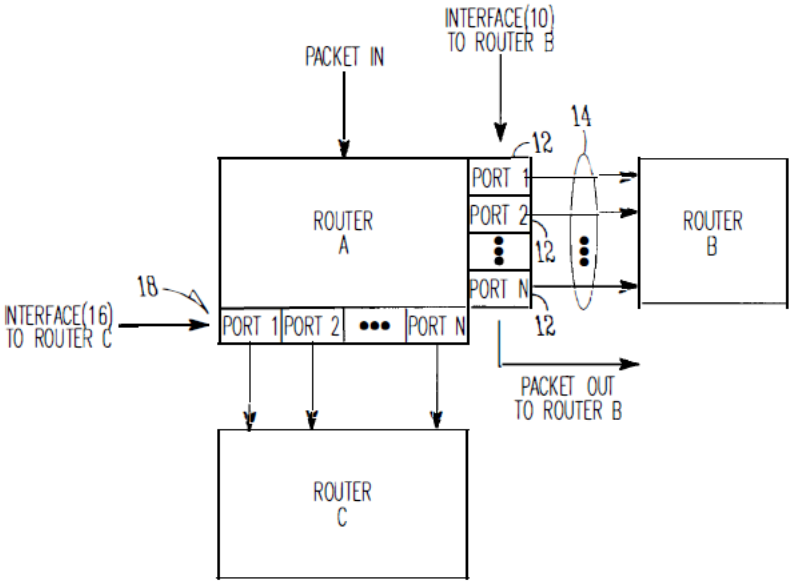
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

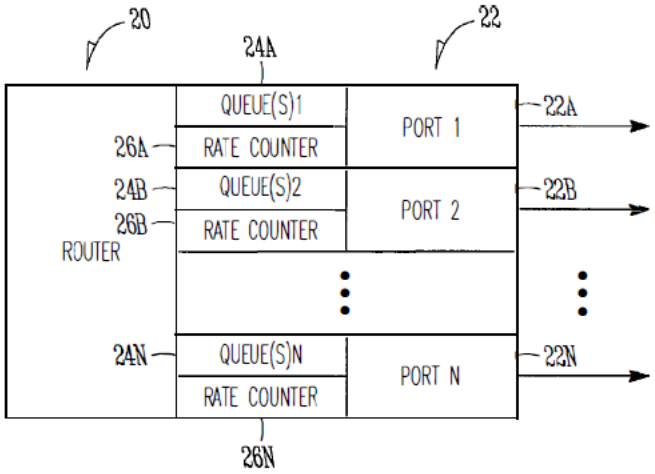
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

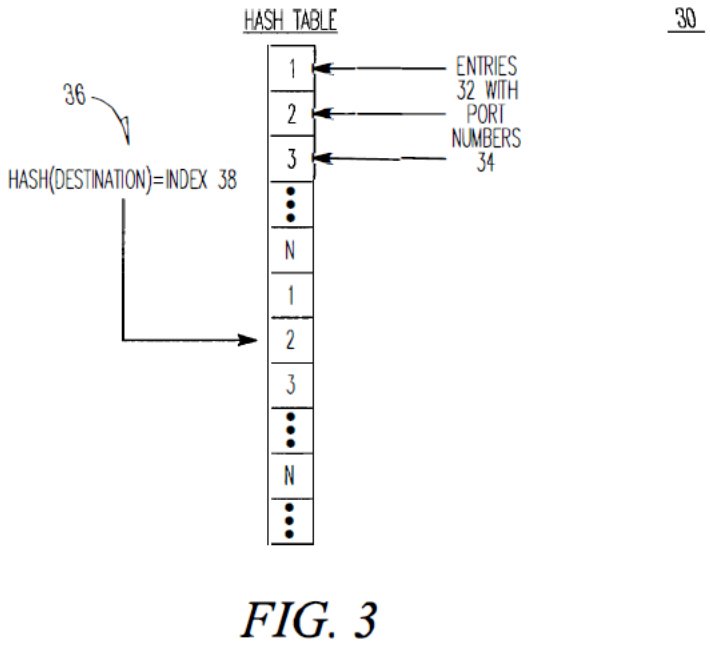
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

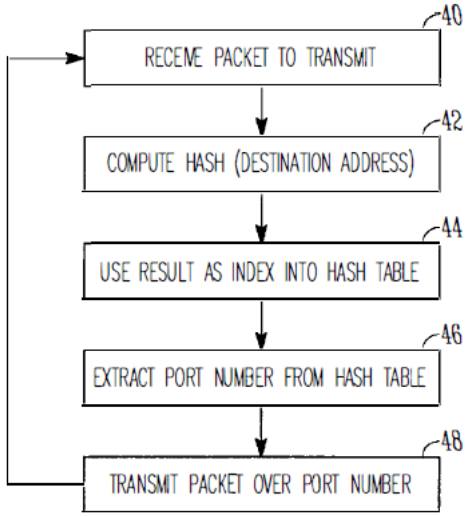
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

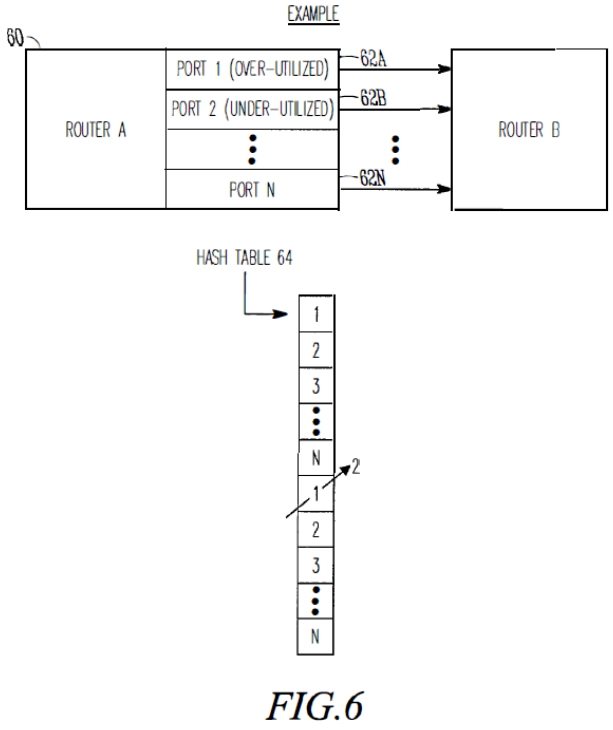
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

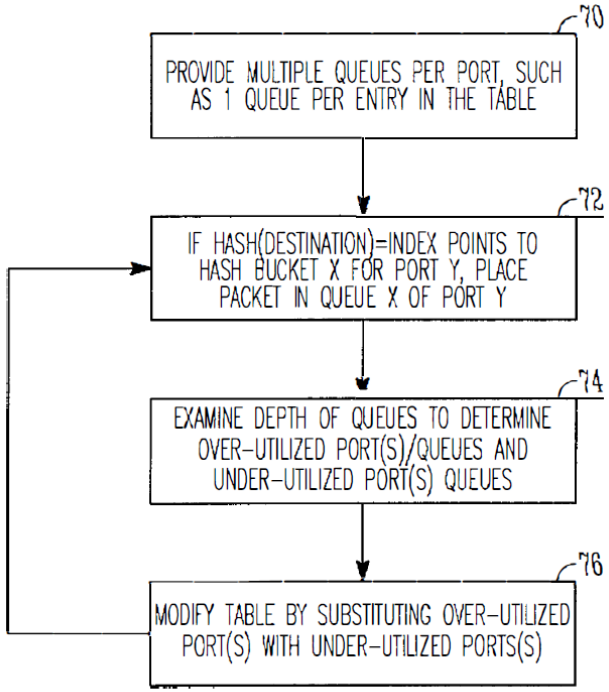
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

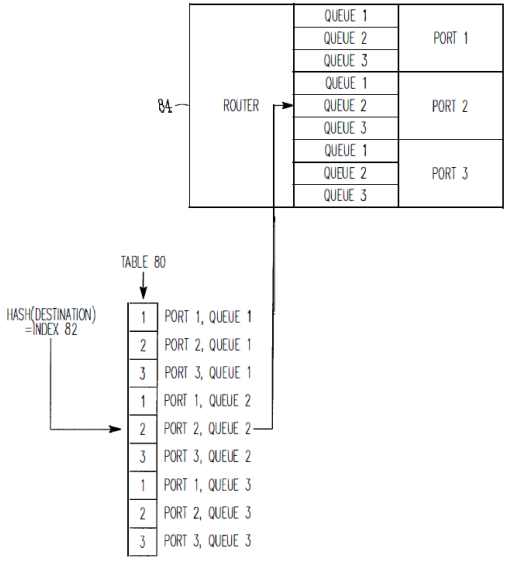
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

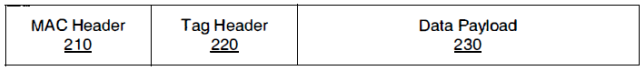
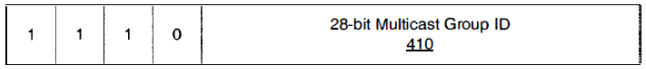
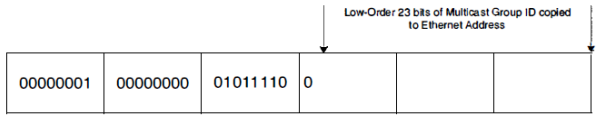
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |
| 30[d] | at least some of the backplane traces are | The Reference discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. |

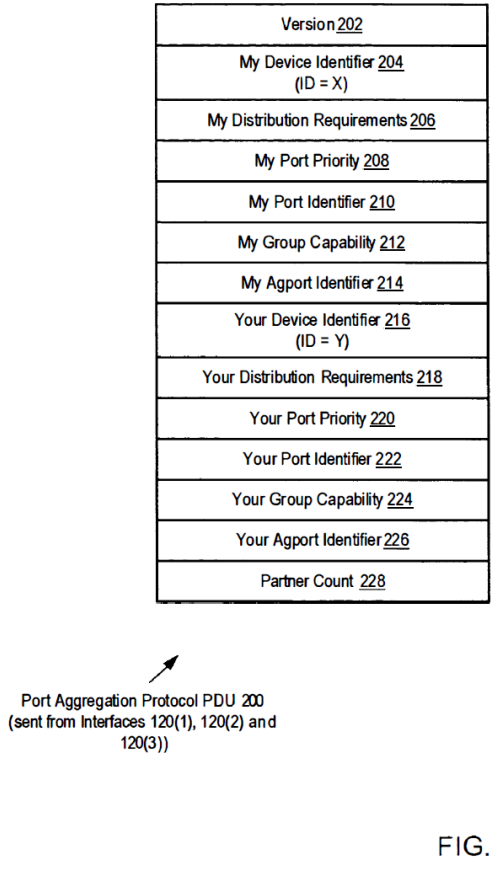
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | aggregated into an Ethernet link aggregation (LAG) group. | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, and Wiher '530, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below is an example.<br><br>Smith '430 at 5:51-64 ("The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be oper-ated as a single logical link, referred to herein as a virtual link bundle. Network device 120(1) operates the two links cou-pling network device 120(1) to virtual network device 202 as 55 a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corre-sponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 60 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, virtual link bundles 250(1) and 250(2) are each operated as an EtherChannel™ or as an aggregated link (as described in IEEE 802.3).")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the link aggregation technique. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125 |

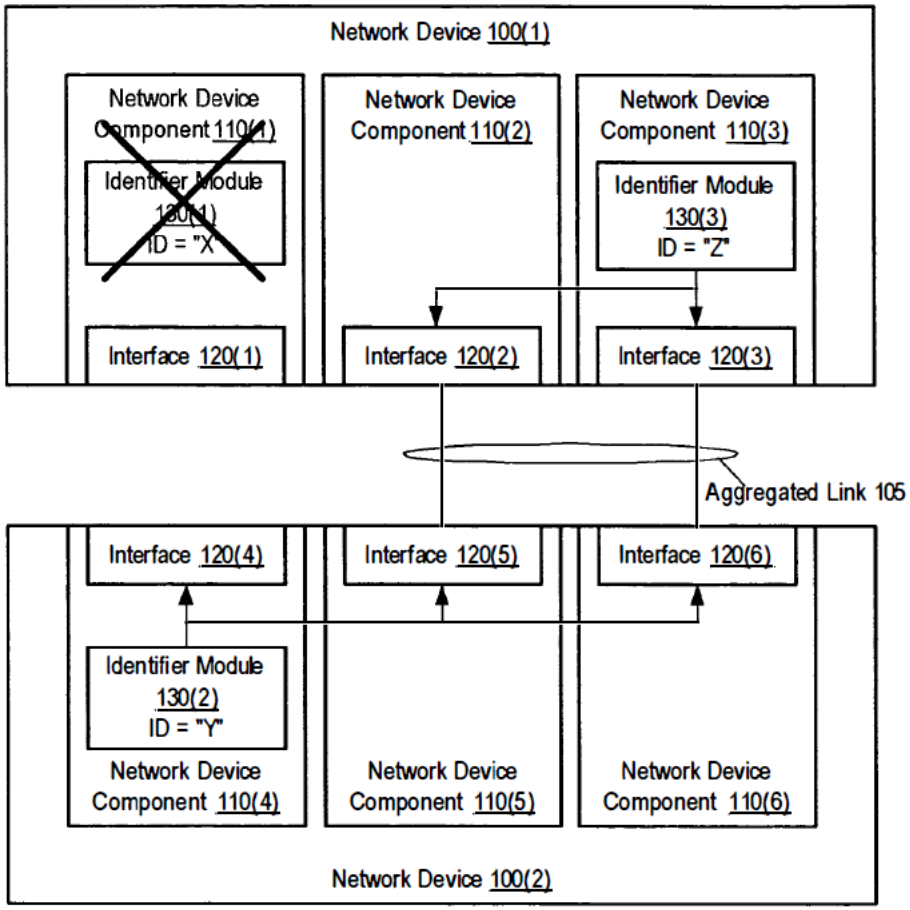
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | DeJager '424 at Abstract ("Provided are methods, apparatuses and systems for balanc-ing the load of data transmissions through a port aggrega-tion. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accom-plished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other net-work environments.")<br><br>DeJager '424 at 1:38-50 ("One way to relieve this bottle-neck is to provide a logical grouping of multiple ports into a single port. The bandwidth of the new port is increased since it has multiple lines (cables) connecting a switch and another network device, each line capable of carrying data at the same rate as the line connecting data sources to the switch. This grouping of ports is sometimes referred to as a port aggregation or port group. One example of such a port aggregation implementation is Cisco Technology, Inc.'s Fast EtherChannel™ port group in a Fast Ethernet network. Further information regarding Fast EtherChannel™ may be found on Cisco Technology, Inc.'s World Wide Web site www.cisco.com. This information is incorporated by reference herein for all purposes.")<br><br>DeJager '424 at 2:47-65 ("The present invention meets this need by providing methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods, apparatuses and systems of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present invention is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utiliza- tion. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by |

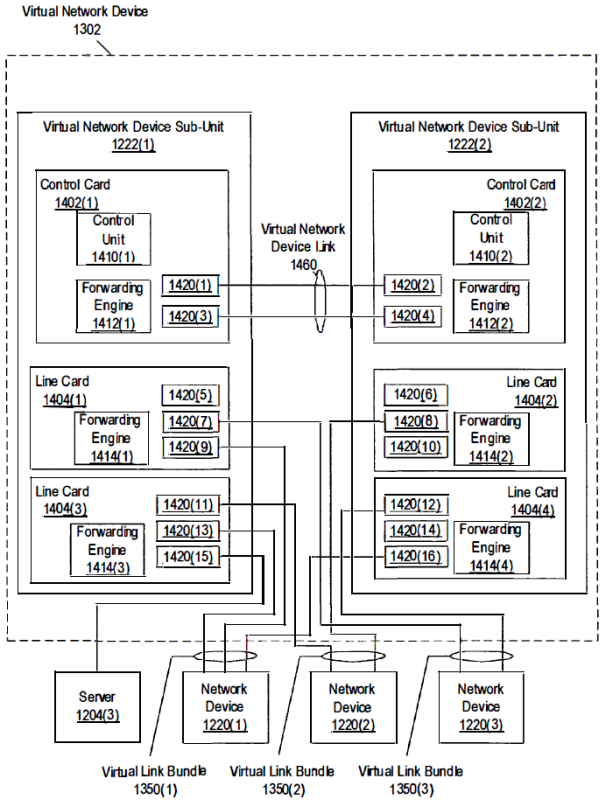
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including l0Base-T, lO0Base-T, and Gigabit Ethernet, and other network envi-ronments.")<br><br>DeJager '424 at 2:67-3:15 ("In one aspect, the present invention provides a method of distributing traffic over a network port group. The method involves receiving a packet of data to be forwarded, deter-mining a stream ID for the packet, and determining whether a prior packet having that stream ID has been distributed to a queue on a port in the group during a predetermined time interval. Where a prior packet having that stream ID has not been distributed to a queue on a port of the group during the predetermined time interval, the method involves allocating the packet to a queue of a port having a lesser load in its queue than a queue of any other port of the group. The method may also involve, where a prior packet having that stream ID has been distributed to a queue on a port of the group during the predetermined time interval, allocating the packet to that queue. In addition, the method may involve monitoring the port group queues to maintain proper iden-tification of the least utilized queue.")<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 4:29-46 ("The present invention provides methods, apparatuses and systems for balancing the load of data transmissions through a port aggregation. The methods and apparatuses of the present invention allocate port assignments based on load, that is, the amount of data being forwarded through each port in the group. The load balancing of the present inven-tion is preferably dynamic, that is, packets from a given stream may be forwarded on different ports depending upon each port's current utilization. When a new port is selected to transmit a particular packet stream, it is done so that the packets cannot be forwarded out of order. This is preferably accomplished by ensuring passage of a period of time sufficient to allow all packets of a given stream to be forwarded by a port before a different port is allocated to transmit packets of the same stream. The invention may be used in a variety of different network environments and speeds, including lOBase-T, lO0Base-T, and Gigabit Ethernet, and other network environments.")<br><br>DeJager '424 at 4:47-58 ("FIG. 1 illustrates a block diagram of a simple network. The network 100 includes two servers S1. and S2, respectively, and two switches, X1 and X2, respectively, as well as four clients C1, C2, C3 and C4, respectively. Clients C , , and C4 are connected to switch X1 by, for example, Fast Ethernet links 102 via ports 1, 2, 3 and 4, respectively. Server S1 is connected to switch X1 via a port aggregation 104, which is a port group composed of ports 5 and 6 of switch X1 . Switch X1 is connected to switch X2 via a second port aggregation 106 which includes ports 7, 8 and 9. Switch X2 is connected to server S2 via port O and Fast Ethernet link 108.")<br><br>Dontu at Abstract ("Various methods and systems for preventing erroneous link aggregation due to component relocation are disclosed. Such methods include a method for changing the identifier used by a network device and communicating the identifier change to a peer network device without disrupting an aggregated link. In one embodiment, a method involves detecting an identifier change and sending a Port Aggrega-tion Protocol (PAgP) protocol data |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | unit (PDU) that includes a new identifier and information. The information indicates the identifier change. The new identifier identifies a network device subsequent to the identifier change. Another embodi-ment of a method involves detecting an identifier change and, subsequent to the identifier change, sending a link aggregation protocol PDU that includes an "old device identifier" field dedicated to conveying an old identifier. The old identifier identifies a network device prior to the iden-tifier change.")<br><br>Dontu at Figure 2 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Dontu at Figure 3 |

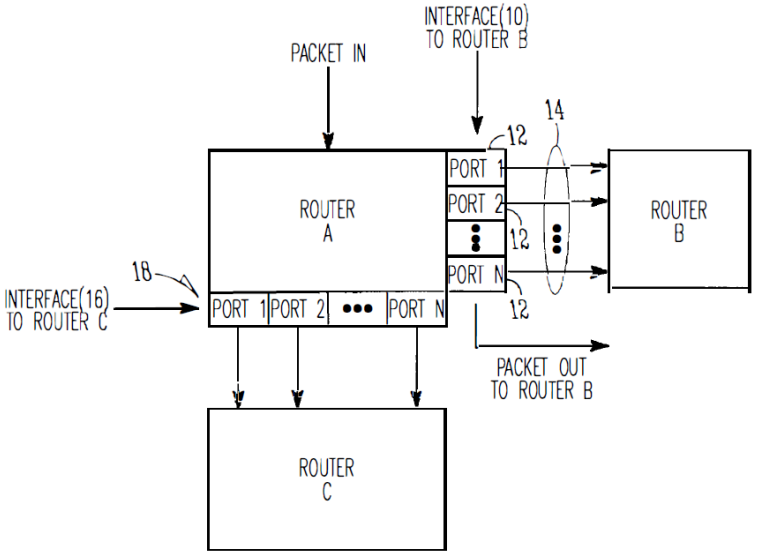
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Dontu at Figure 14 |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>FIG. 14<br><br>Dontu at [0004] ("Link aggregation is used to logically combine two or more individual links into a single aggregated link. Link aggregation can provide improved performance and increased fault tolerance. Improved performance arises because the aggregated link appears to have a bandwidth equal to the combined bandwidth of the individual links. Traffic can be load-balanced among the individual links. Increased fault tolerance is provided since one or more individual links within an aggregated link can fail without disrupting communication between the devices coupled by the aggregated link. Link aggregation techniques include |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Link Aggregation Control Protocol (LACP), which is defined in IEEE 803.2ad, and Port Aggregation Protocol (PAgP), which is a standard promulgated by CISCO SYS-TEMS, INC.")<br><br>Dontu at [0012] ("The method can also involve detecting whether a partner interface is executing a compatible version of PAgP. If the partner interface is not executing the compatible version of PAgP, the compatible version of PAgP can be provided to the partner interface. Alternatively, if the partner interface is not executing the compatible version of PAgP, the partner interface can be inhibited from including a link in an aggregated link.")<br><br>Dontu at [0033] ("Network device 100(1) includes three network device components 110(1)-110(3). Similarly, network device 100(2) includes three network device components 110(4)-110(6). Each network device component 110(1)-110(6) is a component (e.g., a line card, a virtual network device sub-unit (as described below), a chassis useable within a stackable switch, or the like) that can be removed and/or replaced independently of the other network device components. For example, if network device component 110(2) experiences a failure, network device component 110(2) can be removed from network device 100(1) for repair or replacement. The removal of network device component 110(2) does not necessitate the removal of network device components 110(1) and 110(3) from network device 100(1). It is noted that in other embodiments, each network device coupled by an aggregated link can include fewer or additional network device components than the network devices shown in FIG. 1. Additionally, the number of network device components within each network device can vary among network devices (e.g., one network device can include eight network device components, while another network device includes four network device components).")<br><br>Dontu at [0035] ("Aggregated link 105 link includes three links (these links can be physical or logical links). One link couples interface 120(1) to interface 120( 4). Another link couples interface 120(2) to interface 120(5). The third link couples interface 120(3) to interface 120( 6).") |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Dontu at [0037] ("In this example, the network devices 100(1) and 100(2) use Port Aggregation Protocol (PAgP) to form aggre-gated links. Network devices 100(1) each send PAgP pro-tocol data units (PDUs) to each other in order to determine whether any of the links between the two network devices can be combined into an aggregated link. Each PAgP PDU includes an identifier that uniquely identifies the network device that sent that PAgP PDU. Within network device 100(1), identifier module 130(1) of network device compo-nent 110(1) supplies an identifier "X" to each of the inter-faces 120(1)-120(3) within network device 100(1). Inter-faces 120(1)-120(3) include identifier X in each PAgP PDU sent by those interfaces. Similarly, identifier module 130(2) of network device component 110( 4) supplies an identifier "Y" to each interface 120( 4)-120( 6) of network device 100(2). Interfaces 120( 4)-120( 6) include identifier Yin each PAgP PDU sent by those interfaces.")<br><br>Dontu at [0040] ("FIG. 2 illustrates some of the fields that can be included in a PAgP PDU. As shown, PDU 200 includes Version field 202, My Device Identifier field 204 ("My" refers to the device sending the PAgP PDU), My Distribu-tion Requirements field 206, My Port Priority field 208, My Port Identifier field 212, My Group Capability field 212, My Agport (Aggregated Port) Identifier field 214, Your Device Identifier field 216 ("Your" refers to the device to which the PAgP PDU is being sent), Your Distribution Requirements field 218, Your Port Priority field 220, Your Port Identifier field 222, Your Group Capability field 224, Your Agport Identifier field 226, and Partner Count field 228.")<br><br>Dontu at [0110] ("Interfaces 1420(13), 1420(9), and 1420(16), which are each coupled to network device 1220(1) by virtual link bundle 1350(1), form an interface bundle (e.g., an Ether-Channel (TM) port bundle). Similarly, interfaces 1420(11) and 1420(8) form another interface bundle that is coupled to network device 1220(2) by virtual link bundle 1350(2). Interfaces 1420(7) and 1420(12) form a third interface bundle that is coupled to network device 1220(3) by virtual link bundle 1350(3). Within virtual network device 1302, each interface in the same interface bundle is assigned the same logical identifier. For example, interfaces 1420(13), 1420(9), and 1420(16) are each assigned the same logical identifier. In some embodiments, packets received via one of these interfaces are tagged or otherwise associated with the logical identifier to indicate that those packets were received via the |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | virtual link bundle coupling virtual network device 1302 to network device 1220(1). It is noted that similar interface bundles are implemented within each network device 1220(1)-1220(3), and that interfaces included in such bundles are also assigned the same logical identifier by each network device ( or by virtual network device 1302, in embodiments in which virtual network device 1302 controls the configuration of the network devices 1220(1)-1220(3)). For example, network device 1220(1) can assign the same logical identifier to each of the interfaces coupled to virtual link bundle 1350(1).")<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.")<br><br>Li '914 at 2:6-22 ("In light of the above and according to one broad aspect of one embodiment of the present invention, disclosed herein is a method for determining an output port upon which to trans-mit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent router. The method includes creating a list of output ports that are coupled with the adjacent router, modifying the list based on network traf-fic, selecting a port from the list of ports, and transmitting the packet over the selected port. In one example, the list is continuously modified as a background process based on network traffic. The list may be modified by determining a port which is under-utilized, determining a port which is over-utilized, and substituting in the list one or more instances of the port which is over-utilized with one or more instances of the port which is under-utilized. In this manner, the router can adaptively and evenly distribute the packet transmission traffic over the output ports of an interface.") |

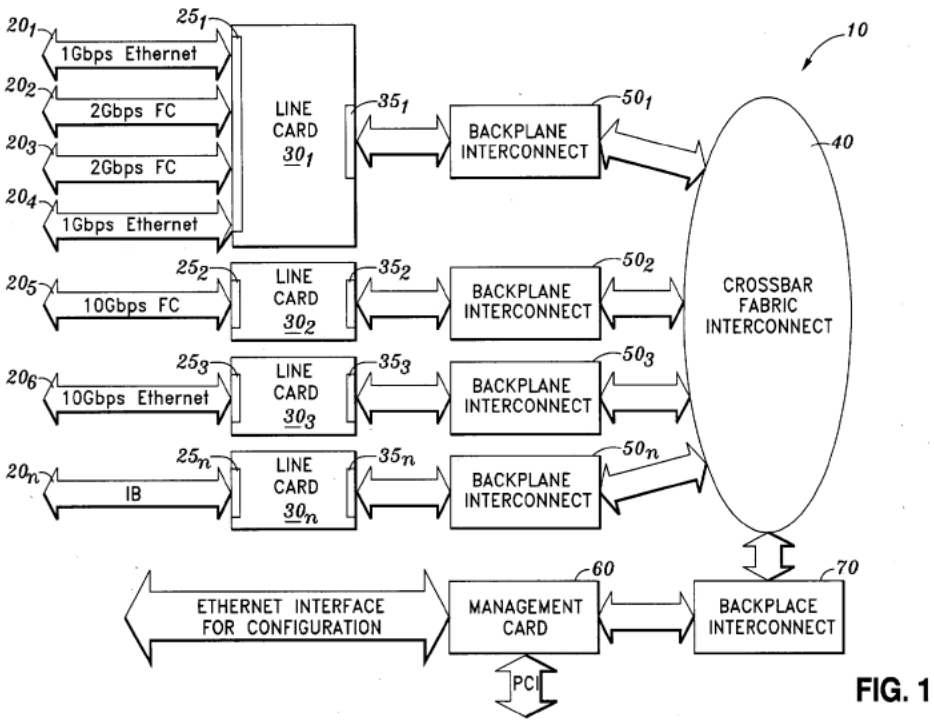
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Li '914 at 4:9-25 ("Referring to FIG. 1, a Router A is shown having an inter-face 10 with a plurality of ports which connect Router A with Router B over a plurality of connections, lines, wires, links or bundled links 14. The ports 12 of Router A are configured to permit transmission of packets from Router A to Router B, and these ports 12 may be referred to as output ports, egress ports, links, or the like. As shown in FIG. 1, port 1 to port N may be connected with Router B, and there may be additional interfaces 16 having ports 18 connected with other routers in the network. When a packet is received by Router A, Router A determines whether the received packet should be transmitted to Router B or to other routers connected to Router A, based in part upon the destination address of the packet. If a packet is to be transmitted from Router A to Router B, then Router A may transmit this packet over the one of the ports 12 shown in FIG. 1.")<br><br>Li '914 at Figure 1<br>  <br><br>*FIG. 1* |

| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | Borgione '125 at 1:55-65 ("Link nodes 110 and 120 can be in physically remote locations, thereby connecting their associated local area networks (LANs). The plurality of network links 150 between link nodes 110 and 120 can be aggregated as a single logical link over which all traffic between link nodes 110 and 120 is distributed. Such aggregation multiplies the available bandwidth for communications between link nodes 110 and 120, and therefore between the two local area networks. When appropriately configured, such a connection can permit the two local area networks to interact as if they were one large local area network.")<br><br>Borgione '125 at 1:66-2:7 ("As stated above, the plurality of network links between 110 and 120 can be aggregated as a single logical link. In this manner, each link node 110 and 120 sees the plurality of network links between them as one logical interface. One type of such an aggregate of links is an EtherChannel, a protocol that allows up to eight Fast Ethernet or Gigabit Ethernet links to be aggregated. Routing protocols treat the aggregated links as a single, routed interface with a common IP address.")<br><br>Borgione '125 at 5:28-50 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even |

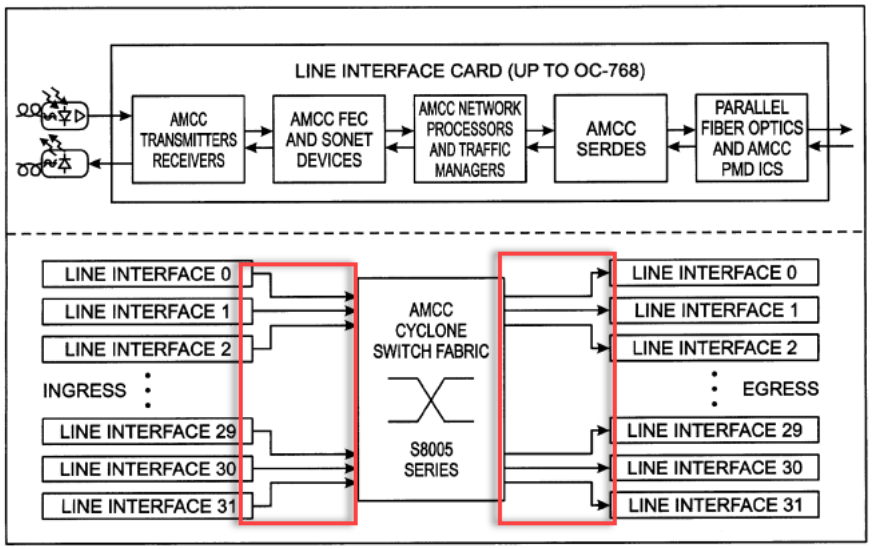
| No. | '740 Patent Claim 30 | The Reference |
|---|---|---|
| | | distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.")<br><br>node are distributed over all com-munication links in the bundle by the load balancing process. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.") |

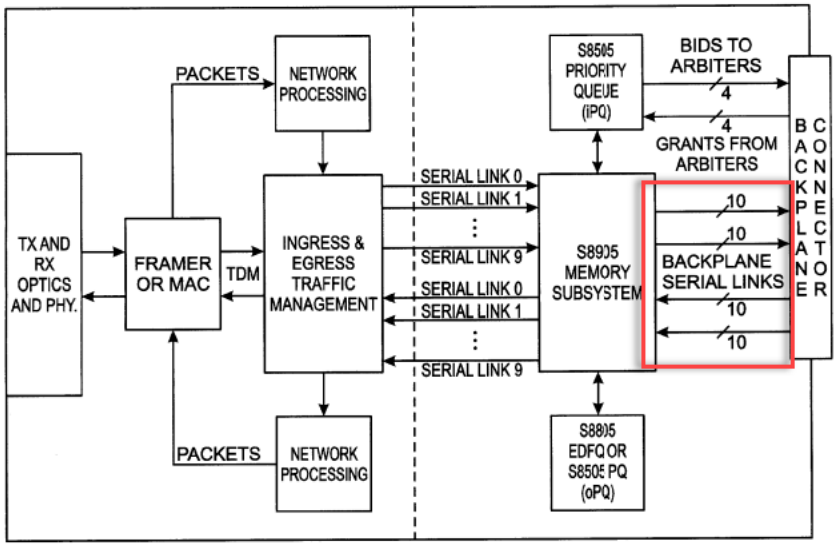
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| 31 | The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. | The Reference discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Hilla, Devi, Cisco EtherChannel System, IEEE 802.3, Cisco EtherSwitch System, Bruckman, Basso, Ghosh, Lebizay, Wiher '530, Viswanathan, Singh, Smith '430, DeJager '424, Dontu, Li '914, and Borgione '125.<br><br>Below are examples of such references.<br><br>Viswanathan at [0028] ("In one embodiment, the invention may be implemented as one or more line cards in a networked environment. To that end, FIG. 1 depicts a simplified schematic of a network interface 10 consistent with the principles of the invention. As shown in FIG. 1, networks 201 -20n ( collectively, "networks 20") are coupled to line interfaces 251-25n ( collectively, "line interfaces 25") of line cards 301 -30n ( collectively, "line cards 30"). Line cards 30 further include fabric interfaces 351-35n ( collectively, "fabric interfaces 35") which serve to couple line cards 30 to crossbar interconnect 40 via backplane |

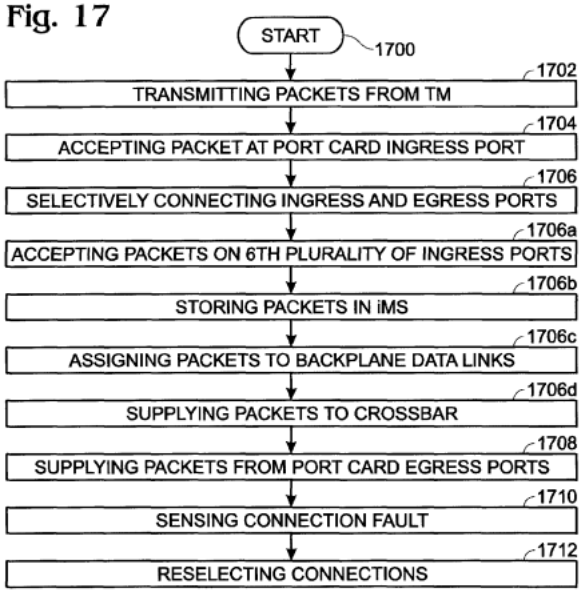
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | interconnects 501-50n (collectively, "backplane interconnects 30"). It should be appreciated that the backplane interconnects 50 may be any switch/ gateway/router capable of connecting line cards 30 to crossbar interconnect 40. Moreover, crossbar interconnect 40 may be used to provide non-arbitrated open communication across all connected systems using a fabric topology (e.g., line cards 30, management card 60, etc.). However, it should equally be appreciated that an arbitrated bus architecture may similarly be used.")<br><br>Viswanathan at [0031] ("Certain management functions for the network interface 10 may be carried out using the management line card 60, which in the embodiment of FIG. 1 is coupled to the crossbar interconnect 40 using backplane interconnect 70. While FIG. 1 depicts only a single Management Line Card 60, it should similarly be appreciated that more than one may be used. In any event, Management Card 60 may execute software for setting up the routing tables for line cards 30, according to one embodiment.")<br><br>Viswanathan at Figure 1 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Singh at 4:5-13 ("In some aspects of the method, selectively connecting port card ingress ports to port card egress ports includes: each port card accepting packets on a plurality of ingress |

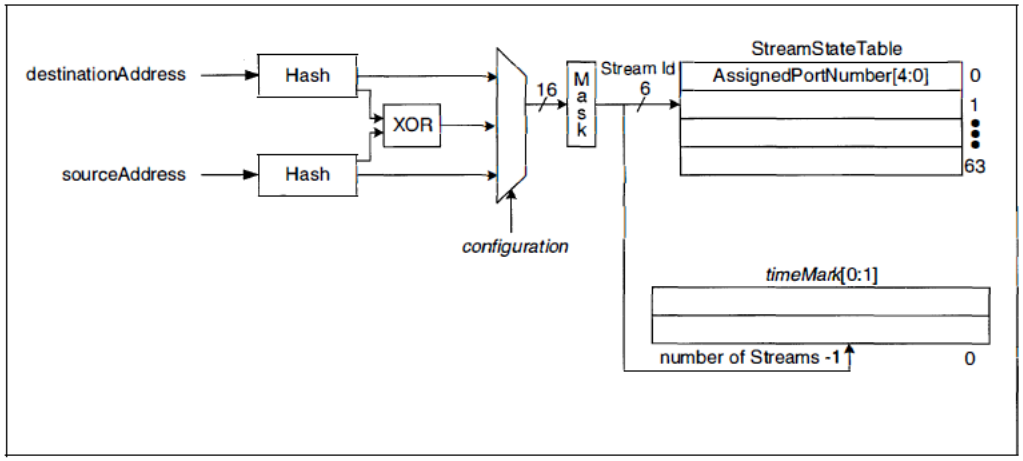
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | data links, through a corresponding plurality of port card ingress ports, from at least one ingress TM (iTM); storing the accepted packets in a port card ingress memory Sub system (iMS); assigning packets to a plurality of port card backplane data links; and, Supplying assigned packets to a crossbar.")<br><br>Singh at 5:54-58 ("A plurality of backplane data links transfers packets between port cards. Shown are a second plurality of ingress 55 backplane data links 1 through k on lines 208 through 210, respectively. Egress backplane data links 1 through t are associated with lines 212 through 214, respectively.)<br><br>Singh at 6:20-29 ("The iPQ 216 has a control link on line 228 operatively connected to a corresponding crossbar 220 controlling the inter-port card transfer of packets on the ingress backplane data links. More specifically, the crossbar is part of a backplane that includes switchplane banks, switchplanes, and switch parts (not shown). Each switch card typically includes a plurality of crossbars controlled by an arbiter that maintains a control link with the iPQ. Additional details of the backplane switching mechanism are provided in Functional Description Section, below.")<br><br>Singh at 13:15-24 ("Two switch cards, that together service a backplane channel, form a switch plane. A backplane channel, as defined in the previous subsection, consists of a group of backplane data links from the MS that carry traffic to the same switch plane. The timing of the links in a backplane channel is such that one link is serviced in the channel every 32 ns with all the links in that channel getting serviced in one cell time. In a fully provisioned 32x32 port card system, there would be 32 4-chamiel port cards and 16 switch cards forming 2 banks of 4 switchplanes as shown in FIG. 11.")<br><br>Singh at Figure 3 (annotations added) |

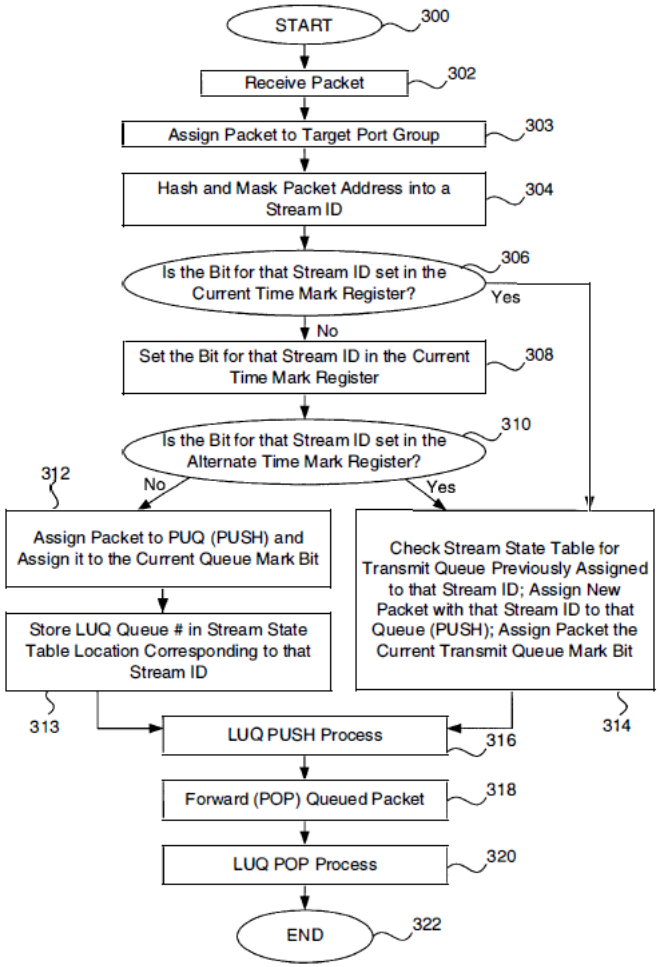
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | Fig. 3<br>  <br><br>Singh at Figure 4 (annotations added) |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | Fig. 4<br>  <br><br>Singh at Figure 17 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | Fig. 17<br>  <br><br>Smith '430 at 9:6-29 ("Thus, providing interconnections between virtual network device sub-units 122(1) and 122(2) can allow virtual network device sub-units 122(1) and 122(2) to operate as a single virtual network device 202. Network devices 120(1)-120(3) communicate with virtual network device 202 in the same way that network devices 120(1 )-120(3) would communicate with a single physical device. For example, if network device 120(2) is handling a packet addressed to server 104(3), network device 120(2) can select one of the two uplinks in network device bundle 250(2) on which to send the packet. This selection can be based on load-sharing criteria. In such a situation, since virtual network device 202 appears to be a single network device, network device 120(2) is just as likely to select the uplink to virtual network device sub-unit 122(2) as the uplink to virtual network device sub-unit 122(1), despite the fact that only virtual network device sub-unit 122(1) has a direct connection to server 104(3). If the packet is sent to virtual network device sub-unit 122(2), network device 122(2) can then use one of the uplinks included in virtual network device link 360 between virtual network device sub-units 122(1) and 122(2) to send the packet to virtual |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | network device sub-unit 122(1), and virtual network device sub-unit 122(1) can in tum provide the packet to its destination, server 104(3).")<br><br>Dontu at [0039] ("Each identifier module 130(1)-130(3) is a part of a network device component that is capable of being the source of a unique identifier. In one embodiment, identifier modules supply media access control (MAC) addresses for use as identifiers. If the network device components are each line cards, the identifier modules can be read-only memories (ROMs) on each of the line cards. The ROMs store the MAC address of each line card. Alternatively, if each network device component is a virtual network device sub-unit, each identifier module can be a backplane. It is noted that other alternatives can be used to supply identifiers such as MAC addresses.")<br><br>Cisco has innovated and patented other improvements to EtherChannel technology, including the use of physical links connecting interface modules to a network node. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• DeJager '424<br>• Dontu<br>• Li '914<br>• Borgione '125<br><br>DeJager '424 at Figure 2 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>DeJager '424 at Figure 3A |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>***FIG. 3A***<br><br>DeJager '424 at 3:16-38 ("In another aspect, the invention provides a network switch. The switch includes a port group and a system for distributing network traffic among ports of the |

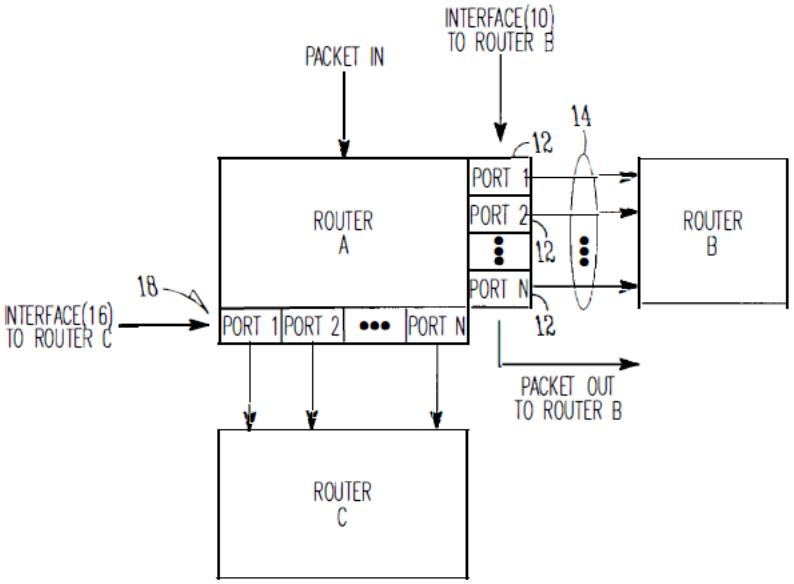
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | port group. The system includes a mechanism for determining a stream ID for the packet and assigning the packet having the stream ID to a queue of a port in the port group, and a mechanism for adjusting a queue assignment of a prior packet having the stream ID to a queue of a different port of the port group based on load in the queues of the ports of the group. The mechanism for determining a stream ID and assigning the packet having the stream ID to a queue of a port in the port group may include a hashing and masking mechanism for determining a stream ID for the packet, a pair of time mark registers for determining whether another packet having the stream ID has been distributed to a queue for a port in the group during a time interval, and a stream state table for storing stream IDs with corresponding queue assignments. The adjusting mechanism may include a least utilized queue register for maintaining proper identification of a least utilized queue, and a pair of queue mark registers for determining whether a queue for a port in the port group is current. In addition, the switch may include a clock for timing a load balance time interval.")<br><br>DeJager '424 at 5:19-30 ("Ethernet addresses have 48 bits. Therefore, the number of possible streams identified by such an address may be 248 or, where the stream address is defined by both the source and the destination address, 296. In order to reduce the number of possible stream addresses and thereby permit a more economical system, both addresses may be hashed and then either an XOR (exclusive OR logical operation) of the two hashes or one of the hashes independently may masked down to a n-bit index, where n is much less than 96, for example 6, as shown in FIG. 2. Conventional hashing and masking techniques and mechanisms known to those of skill in the art may be used. This results in a table depth of 64 (26).")<br><br>DeJager '424 at 5:42-45 ("Once it has been hashed and masked, the 6-bit stream identification (stream ID) is used to address the stream state table. This table stores the port number currently assigned to a stream.")<br><br>DeJager '424 at 7:59-8:10 ("FIG. 3Ais primarily addressed to the basic load balancing feature of the present invention, that is, assignment of packets to ports in a port aggregation based on traffic volume. The process begins at a step 300, and at a step 302 a packet of data |

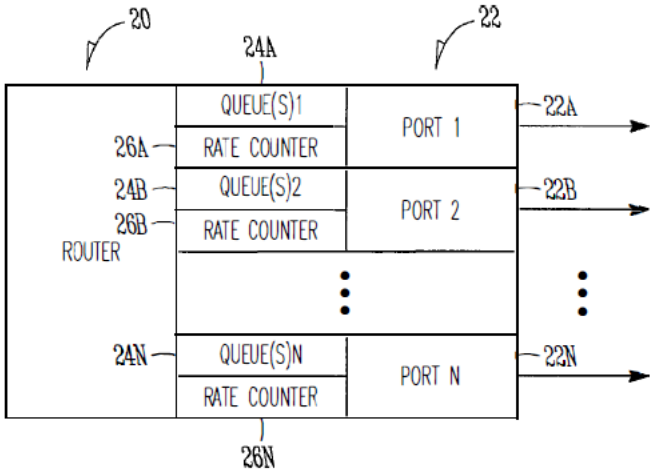
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | is received for forwarding. At a step 303 the packet's address is analyzed to determine the packet's target port group. At a step 304, the packet's address is hashed and masked into a 6-bit stream ID. Next, in a decision step 306, a determination is made whether or not the bit for that stream ID is set in the current time mark register. If decision step 306 is answered in the negative, the bit for that stream ID is set in the current time mark register, at a step 308. If decision step 306 is answered in the affirmative, the stream state table is checked for the transmit queue to which the packets from the stream corresponding to that stream ID have previously been assigned, and the new packet having the same stream ID is assigned to (pushed onto) that transmit queue, at a step 314. The newly queued packet is also assigned the current transmit queue mark bit.")<br><br>DeJager '424 at 9:17-26 ("FIG. 4 shows a block diagram of a load balancing system in accordance with a preferred embodiment of the present invention. In this embodiment, the system 400 includes a "switch" 402, which may be a switch or other packet-forwarding device as described previously, to which inbound links 401 from sources in the network transmitting packets are connected. The incoming packets pass through a mechanism 404 for hashing and masking packet addresses in order to assign each packet an appropriate stream ID, for example as described previously.")<br><br>Dontu at [0095] ("In some embodiments, network devices 1220(1) and 1220(2) are aware (e.g., through various state informa-tion maintained within each network device) that each virtual link bundle 1350(1) and 1350(2) includes links that are terminated on different network devices in distribution layer 1212. In such an embodiment, network devices 1220(1) and 1220(2) can select a link within a particular virtual link bundle on which to send a packet based on this awareness.")<br><br>Dontu at [0097] ("FIG. 13B illustrates another embodiment of the present invention. In FIG. 13B, network devices 1220(1) and 1220(2) operate in the same manner that those network devices would operate if connected to a single network device. By operating in this manner, the use of a virtual link bundle is simplified. For example, if network device 1220(1) is aware |

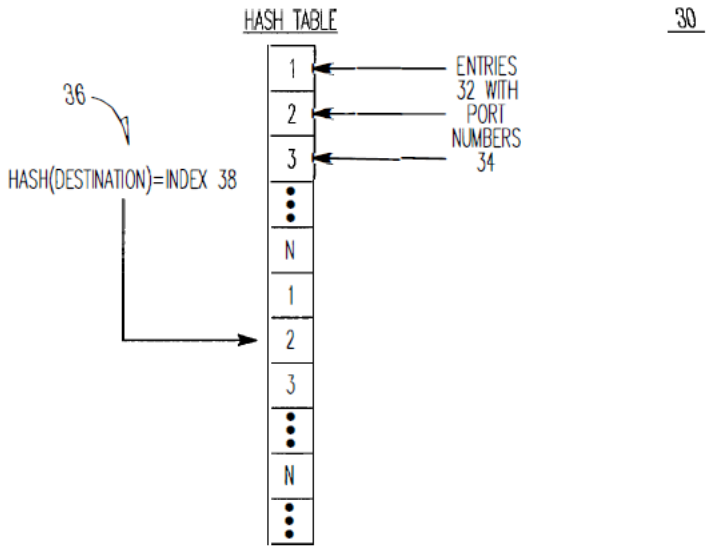
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | that virtual link bundle 1350(1) terminates at two different network devices, network device 1220(1) selects a link on which to send a particular packet based on Spanning Tree Protocol. The use of Spanning Tree Protocol may involve more overhead and/or be more restrictive with respect to which links can be used to send a given packet (e.g., Spanning Tree Protocol might block all but one of the links, preventing utilization of all but one non-blocked link) than if network device 1220(1) simply views virtual network device 1302 as a single entity. When viewing virtual net-work device 1302 as a single entity, for example, network device 1220(1) simply select a link on which to send a packet based on load-sharing constraints. Similarly, if a link within virtual link bundle 1350(1) fails, there is no need for network device 1220(1) to change how Spanning Tree Protocol is applied. Instead, network device 1220(1) simply continues to use the non-failed links within virtual link bundle 1350(1).")<br><br>Dontu at [0108] ("Thus, providing interconnections between virtual network device sub-units 1222(1) and 1222(2) allows virtual network device sub-units 1222(1) and 1222(2) to operate as a single virtual network device 1302. Network devices 1220(1)-1220(3) communicate with virtual network device 1302 in the same way that network devices 1220(1)-1220(3) would communicate with a single physical device. For example, if network device 1220(2) is handling a packet addressed to server 1204(3), network device 1220(2) selects one of the two uplinks in network device bundle 1350(2) on which to send the packet. This selection is based on load-sharing criteria in some embodiments. In such a situation, since virtual network device 1302 appears to be a single network device, network device 1220(2) is just as likely to select the uplink to virtual network device sub-unit 1222(2) as the uplink to virtual network device sub-unit 1222(1), despite the fact that only virtual network device sub-unit 1222(1) has a direct connection to server 1204(3). If the packet is sent to virtual network device sub-unit 1222(2), network device 1222(2) uses one of the uplinks included in virtual network device link 1460 between virtual network device sub-units 1222(1) and 1222(2) to send the packet to virtual network device sub-unit 1222(1), and virtual network device sub-unit 1222(1) can in turn provide the packet to the packet's destination, server 1204(3).")<br><br>Dontu at [0109] ("In other embodiments, network devices 1220(1)-1220(3) are aware that virtual link bundles 1350(1) and 1350(2) actually terminate on two different network devices. |

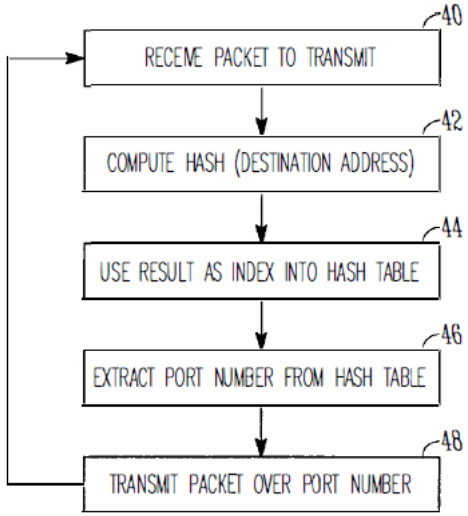
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | Network devices 1220(1)-1220(3) control packet transmis-sion based on this information. For example, in this situa-tion, network device 1220(2) handles a packet addressed to server 1204(3) by selecting the uplink coupled to virtual network device sub-unit 1222(1) instead of the uplink coupled to virtual network device sub-unit 1222(2), based on the fact that network device 1220(2) recognizes separate connections to two different network devices within the logical link.")<br><br>Dontu at [0112] ("The same logical identifiers are used to identify uplink interface bundles by each of virtual network device sub-units 1222(1) and 1222(2), and the virtual network device sub-units coordinate to assign the same logical iden-tifier to each uplink interface within the same uplink inter-face bundle. When forwarding packets via an uplink inter-face bundle identified by a particular logical identifier, each virtual network device sub-unit 1222(1) and 1222(2) gen-erates a hash value to select one of the uplink interfaces within that uplink interface bundle on which to send the packet. Each of the virtual network device sub-units uses these hash values to identify local uplink interfaces within that virtual network. Thus, each virtual network device sub-unit will only select an uplink interface that is local to that virtual network device sub-unit. For example, if virtual network device sub-unit 1222(1) is forwarding a packet via the uplink interface bundle that includes interfaces 1420(9), 1420(13), and 1420(16), the hash value generated by virtual network device sub-unit will identify one of interfaces 1420(9) or 1420(13).")<br><br>Dontu at [0113] ("In the above example, by associating each hash value with local uplink interfaces in the uplink interface bundle, the usage of virtual switch link 1460 is reduced. Essentially, virtual network device sub-unit 1222(1) favors local uplink interfaces within a particular uplink interface bundle over remote uplink interfaces, in the same uplink interface bundle, on virtual network device sub-unit 1222(2). Likewise, virtual network device sub-unit 1222(2) favors local uplink interfaces within a particular uplink interface bundle over uplink interfaces included in virtual network device sub-unit 1222(1). For example, if virtual network device sub-unit 1222(2) needs to forward a packet via an uplink interface, virtual network device sub-unit 1222(2) will send that packet via uplink interface 1420(12) instead of forwarding that packet across virtual network device link 1460 to be sent via uplink |

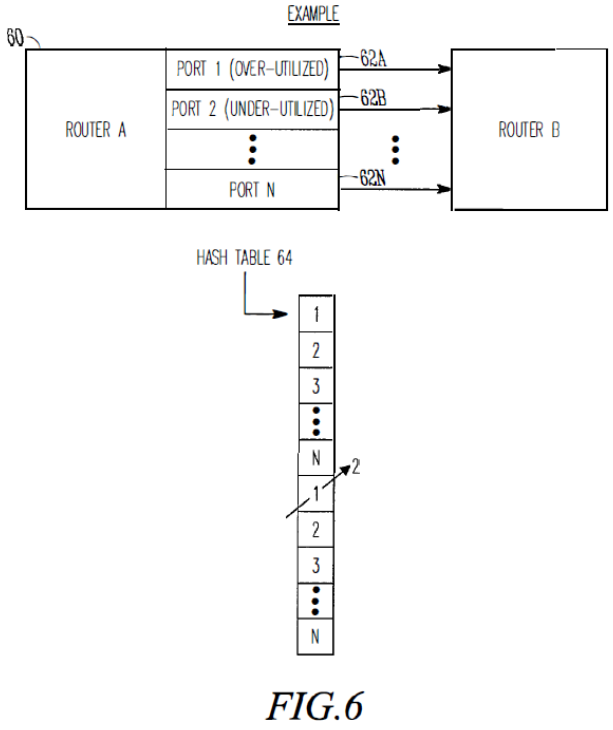
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | interface 1420(7). By favoring local interfaces, the amount of traffic sent over virtual network device link 1460 is reduced, since each virtual network device sub-unit 1222(1) and 1222(2) will forward locally-received packets (i.e., packets received via interfaces other than those coupled to virtual network device link 1460) from a local interface.")<br><br>Dontu at [0118] ("To operate in this way, each egress uplink interface coupled to a link in a virtual link bundle is configured to filter out traffic received via virtual network device link 1460. For example, a packet is received at virtual network device sub-unit 1222(1) via virtual network device link 1460. The interface 1420(1) or 1420(3) that receives the packet updates information (e.g., in a header) associated with the packet to indicate that the packet was received via virtual network device link 1460 (in alternative embodi-ments, the sending interface in virtual network device sub-unit 1222(2) can update this information). When virtual network device sub-unit 1222(1) looks up the destination address of the packet in a lookup table, the lookup table returns the logical identifier that identifies local uplink interfaces 1420(9) and 1420(13). The packet is then for-warded to uplink interface 1420(13) (e.g., selected based on load-sharing considerations). When uplink interface 1420(13) receives the packet, uplink interface 1420(13) will only output the packet if the packet was not received via virtual switch link 1460, since if the packet was received via the virtual switch link, the other virtual network device sub-unit 1222(2) will have already sent the packet via the virtual link bundle. Thus, uplink interface 1420(13) can filter the packet from the packet flow being sent via uplink interface 1420(13) based on the information appended to the packet that indicates whether the packet was received via virtual network device link 1460.")<br><br>Li '914 at Figure 1 |

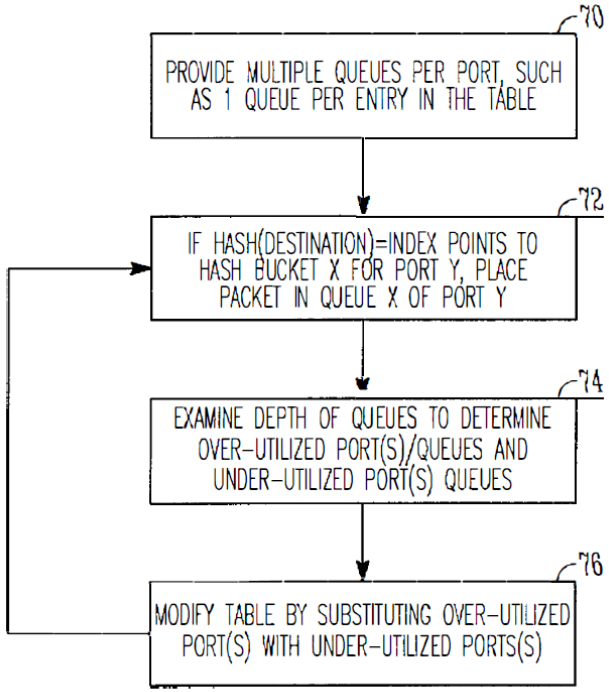
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Li '914 at Figure 2 |

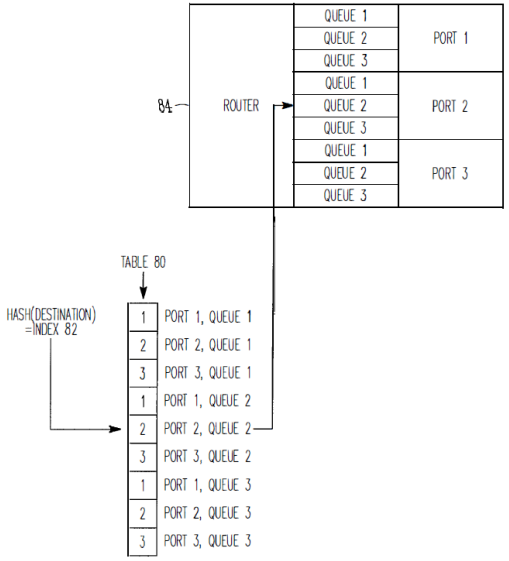
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 2*<br>Li '914 at Figure 3 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br>Li '914 at Figure 4 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Li '914 at Figure 6 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG.6*<br>Li '914 at Figure 7 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 7*<br>Li '914 at Figure 8 |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 8*<br><br>Li '914 at 1:30-43 ("For instance, in the example of FIG. 1, Router A has an interface 10 with a plurality of ports or links 12 which connect with Router B in order to pass data from Router A to Router B, in this example. When a packet from Router A needs to be transmitted to Router B, Router A determines which port of the plurality of ports 12 should be used to transmit the packet to Router B. Conventionally, a hash operation or function may be used to generate an index into the plurality of ports of Router A. For instance, a hashing function may be performed using the destination address of the packet to generate an index, and unneeded bits may be masked off in order to form an index which is used to select one of the plurality of ports 12 of the interface 10 of Router A upon which to transmit the packet.") |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | Li '914 at 1:44-57 ("As recognized by the present inventors, such a process is a static process which is not sensitive to the amount of traffic being handled by particular ports 12 of Router A. In other words, if an amount of traffic builds up on one or more ports 12 of Router A, the above-described hash function does not account for such traffic build-ups in determining which port of interface 10 should be utilized to transmit a packet. As recognized by the present inventors, this problem may be compounded when adjacent routers are of the same make and model and use the same hashing function, such that a build-up of traffic on a particular port in Router A may be propagated and compounded onto a corresponding port of Router B, which degrades the overall performance of Router A and Router B in the network.")<br><br>Li '914 at 2:39-55 ("In one example, the operation of selecting a port from the list of ports may include performing a hash operation using a destination address of the packet to generate an index value into the list, and selecting a port from the list based on the index value.<br>According to another broad aspect of another embodiment of the invention, disclosed herein is a method for determining an output port upon which to transmit a packet in a router having a plurality of output ports adapted to be coupled with an adjacent or "next-hop" router. The method includes creat-ing a list of output ports that are coupled with the adjacent router; updating the list based on network traffic over the output ports; extracting a destination address from the packet; performing a hash function using the destination address to create an index into the list; at the location of the index in the list, extracting an identifier of an output port; and transmitting the packet over the output port.")<br><br>Li '914 at 4:41-67 ("In FIG. 3, a data structure or table 30 such as a hash table is shown, in accordance with one embodiment of the present invention. In the example of FIG. 3, the table 30 includes a plurality of entries 32 which in one embodiment, are filled with the port numbers 34 of the ports of the router. In this example, assuming that there are N ports of Router A which are coupled with Router B, then the table contains as entries 32 the port numbers 1 to N. The size of the table is a matter of choice, and in one example, contains 65,536 entries to support 16-bit addressing. |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | In one example, the destination address 36 of the packet to be transmitted is used as the operand of a hash operation. In general, a hashing function is an operation which produces a unique numeric value based upon a given operand. The result of this operation is used as an index 38 into the table 30 shown in FIG. 3, and the port number 34 contained within the entry 32 indexed is utilized to transmit the packet out of the router. In one example, the hash operation generates a 16-bit result, and in this example, the table 30 is sized to support 65,536 entries. In one example, the table 30 is filled with port num- bers 34 in a sequential manner, such as shown in FIG. 3. It is understood that the length of the table 30 is a matter of choice depending upon the particular implementation. Further, it is understood that while a 16-bit result from the hash operation may be used in on example, a portion of the 16-bit result may be masked off to form a result of less than 16 bits, if desired, or a larger address range may be used.")<br><br>Li '914 at 5:13-28 ("In FIG. 4 at operation 40, a packet is to be transmitted to a particular adjacent router for "next hop" in the network over one or more ports of the router. For example, in FIG. 1, a packet is to be transmitted from Router A to Router B over one of the plurality of ports 12. In FIG. 4, at operation 42, a hash operation is performed using, in one example, the destination address of the packet. The result of the computation is used at operation 44 as an index into a table or data structure, such as the hash table 30 shown in the example of FIG. 3. In FIG. 4, at operation 46, a port number is extracted from the entry of the table indexed by operation 44. At operation 48, the packet is transmitted from the router along the ports identified by the port number extracted by operation 46. Operations 40-48, or various combinations thereof, may be repeated as needed to handle the transmission of multiple packets over the various ports between routers.")<br><br>Li '914 at 5:29-41 ("Referring to the example of FIG. 3, assuming that a packet has a destination address 36 which, upon performing a hash function yields an index 38 which points to the second entry in the table 30, the port number "2" is extracted from the table, and the packet is transmitted to the adjacent router in the network over port number 2, in this example. If another packet to be transmitted had a destination address 36 which, upon performing a hash operation, generates an index 38 pointing to the first entry in the hash table |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | 30, then in this example the port number "1" is extracted from the table and the packet is transmitted to appropriate adjacent router using port number "1." These examples are provided for purposes of illustration only.")<br><br>Li '914 at 5:66-6:8 ("Upon determining the overutilized and underutilized ports of the router, operation 54 modifies the table 30 by substitut- ing an overutilized port with an underutilized port. In one example, where a hash table 30 contains multiple entries having a port listed multiple times within the table, a single substitution of one instance of an overutilized port is made using an underutilized port. In this manner, the changes in the traffic between the overutilized and underutilized ports are made at a low rate so that the traffic is smoothly distributed across the ports.")<br><br>Li '914 at 6:65-7:7 ("At operation 72 of FIG. 7, a hash operation is performed using, in one example, the destination address of the packet to be transmitted. The result of the hash operation generates an index into the table, and if the index points to an entry corre-sponding to a particular port, then the packet is placed in the corresponding queue of the particular port. For example and referring to FIG. 8, if the hash operation generates an index 82 which points to the fifth entry in the table 80 (shown as corresponding to port 2, queue 2), then the packet is placed in queue 2 of port 2 for transmission out of the router 84.")<br><br>Borgione '125 at 2:8-18 ("Load balancing of data packets transmitted across individual network links within an aggregate of network links can be handled by interface hardware. The individual network links, across which the data load is to be balanced, can be selected in several ways. One such way is to analyze source and destination Ethernet addresses within the data packets to be sent over the logical link and generate a link identifier from that information. Another method for selecting a network link over which to send a packet is a round robin method, wherein each link is selected in order as packets arrive.")<br><br>Borgione '125 at 3:14-23 ("A multicast packet is typically transmitted as a single packet received by a select group of receivers. The group of receivers is designated by a multicast address. The source node address appears in the header of a multicast packet, and the |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | multicast address appears as the destination address. A single multicast packet sent by a network node can be repli-cated at other network nodes, such as link nodes 110 and 120, in order for the receivers to receiver the multicast packet. Each replicated multicast packet will have the same source and destination address (the multicast address).")<br><br>Borgione '125 at 3:30-49 ("As stated above, packet source and destination addresses can be analyzed to determine which network link in a logical link is to be used to send a packet between link nodes 110 and 120. Commonly, such analysis involves a hashing algorithm that takes the Ethernet addresses and generates a network link identifier. The network link identifier identifies which of the plurality of network links is to be used for sending the packet between link nodes 110 and 120.<br>While the aforementioned method addresses data load bal-ancing for certain types of data transmission (e.g., unicast), the method does not efficiently balance data loads across individual network links within a logical link for more com-plex data transmission such as multicast packet transmission. To illustrate, if a multicast packet is replicated at a link node (e.g., link node 110 or 120), the source and destination address are the same for replicated multicast packets, and such a hashing algorithm will generate the same link identifier for each replicated multicast packet and therefore send all of those replicated multicast packets on the same network link. This can create an undesirable load imbalance among the plurality of network links.")<br><br>Borgione '125 at 4:3-30 ("Accordingly, one aspect of the present invention provides a method for transmitting a replicated multicast packet over one of a plurality of network links that form one logical channel. Selecting the one of the plurality of network links comprises analyzing a destination ethernet address of the replicated multicast packet and a non-ethernet component of the header of the replicated multicast packet.<br>A further aspect of the present invention provides a method for replicating a multicast packet to produce first and second multicast packets, which are transmitted over a first and sec-ond link of a logical channel between a pair of network nodes.<br>Another aspect of the present invention provides a system comprising a first network node coupled to a second network node through a plurality of network links. The first network |

| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | node selects a destination interface identifier for an outgoing multicast packet, selects one of the plurality of network links using the destination interface identifier, and transmits the outgoing multicast packet to the second network node over the selected network link.<br>Another aspect of the present invention provides a method comprising connecting a first network device to a second network device using a plurality of network links. A multicast packet is provided to the first network device, which is con-figured to replicate the multicast packet thus forming repli-cated multicast packets. Each replicated multicast packet receives a destination interface identifier which is used to select one of the plurality of network links for transmitting the replicated multicast packet by the first network device.")<br><br>Borgione '125 at 5:28-54 ("The present invention balances the transmission of repli-cated multicast packets among an aggregate of network links that provide a logical channel or link between network nodes. Prior art link load balancing requires analysis of source and destination Ethernet addresses (i.e., as input to a hashing algorithm). Since replicated multicast packets each have the same source and destination Ethernet addresses, another part of a replicated multicast Ethernet packet must be used in order to differentiate between replicated multicast Ethernet pack-ets. An added tag header can be used to include a destination interface identifier. For example, in a YLAN network envi-ronment, such a tag header is included in packets per IEEE Std. 802.lQ. A portion of an IEEE Std. 802.lQ tag header is a YLAN identifier (YID), which is unique to a particular YLAN. A destination interface identifier within a tag header can be used to select which network link in a logical link is to be used to transmit a replicated multicast packet. Since the destination interface identifier often varies from replicated multicast packet to replicated multicast packet, use of the destination interface identifier to select a network link will lead to a more even distribution of multicast packet transmis-sion across the logical link. Such a distribution can reduce the likelihood of a load imbalance in the logical link.<br>Network packets contain header information and data pay-load information. Header information can include Media Access Control (MAC) addressing such as the source and destination addresses of the packet.") |

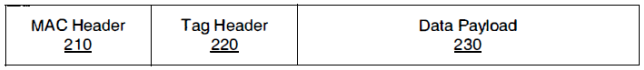
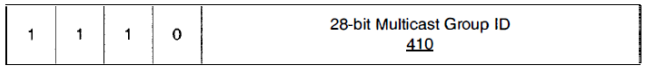
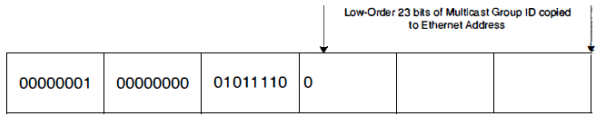
| No. | '740 Patent Claim 31 | The Reference |
|---|---|---|
| | | Borgione '125 at 7:1-5 ("A calculation that takes place in step 670 can take any form that generates an output value from an input value. A hash algorithm is one form of such a function. A hash function can have as an input a destination interface identifier (such as YID).")<br><br>Borgione '125 at Figure 2-5<br>  <br><br>**Figure 2**<br>  <br><br>**Figure 3**<br>  <br><br>**Figure 4**<br>  <br><br>**Figure 5** |

**EXHIBIT E-4**
Defendant's First Amended Invalidity Contentions
*Orckit Corporation v. Cisco Systems, Inc.*, 2:22-cv-00276-JRG-RSP

---

**Chart for U.S. Patent 10,652,111 ("the '111 Patent")**
**35 U.S.C. §103**

In this chart, "Reference" refers to each of the following the References:

- U.S. Patent Publication No. 2012/0300615 to Kempf et al. ("Kempf")
- U.S. Patent Publication No. 2013/0322242 to Swenson et al. ("Swenson")
- U.S. Patent Publication No. 2014/0140211 to Chandrasekaran et al. ("Chandrasekaran")
- U.S. Patent No. 9,264,400 to Lin et al. ("Lin '400")
- U.S. Patent Publication No. 2013/0291088 to Shieh et al. ("Shieh '088")
- Cisco Intelligent WAN ("Cisco IWAN System")
- VMware NSX Network Virtualization ("VMware NSX System")
- U.S. Patent No. 9,276,877 to Chua '877 et al. ("Chua '877")
- U.S. Patent No. 9,038,151 to Chua '151 et al. ("Chua '151")
- U.S. Patent Publication No. 2005/0210533 to Copeland et al. ("Copeland")
- U.S. Publication No. 2011/0310901 A1 to Uchida et al. ("Uchida")

The following references are identified individually:

- OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04) ("OpenFlow")
- U.S. Patent Publication No. 2006/0083167 to Balakrishnan ("Balakrishnan")
- U.S. Patent No. 9,467,478 to Khan et al. ("Khan '478")
- U.S. Patent No. 8,868,735 to Wang et al. ("Wang '735")
- U.S. Patent No. 10,142,254 to Olofsson et al. ("Olofsson '254")
- U.S. Patent No. 9,614,739 to Kumar et al. ("Kumar '739")

As shown in the chart below, all Asserted Claims of the '111 Patent are invalid under 35 U.S.C. § 103 because The Reference renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further

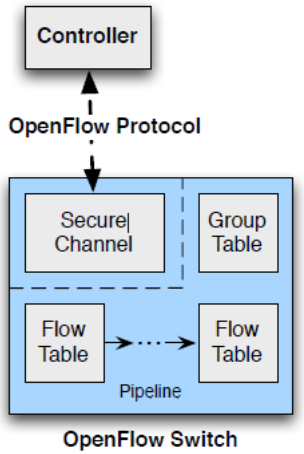
combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's First Amended Invalidity Contentions.

Motivations to combine include at least the similarity in subject matter between the references to the extent they concern methods relating to routing certain network traffic to entities for further analysis and inspection. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

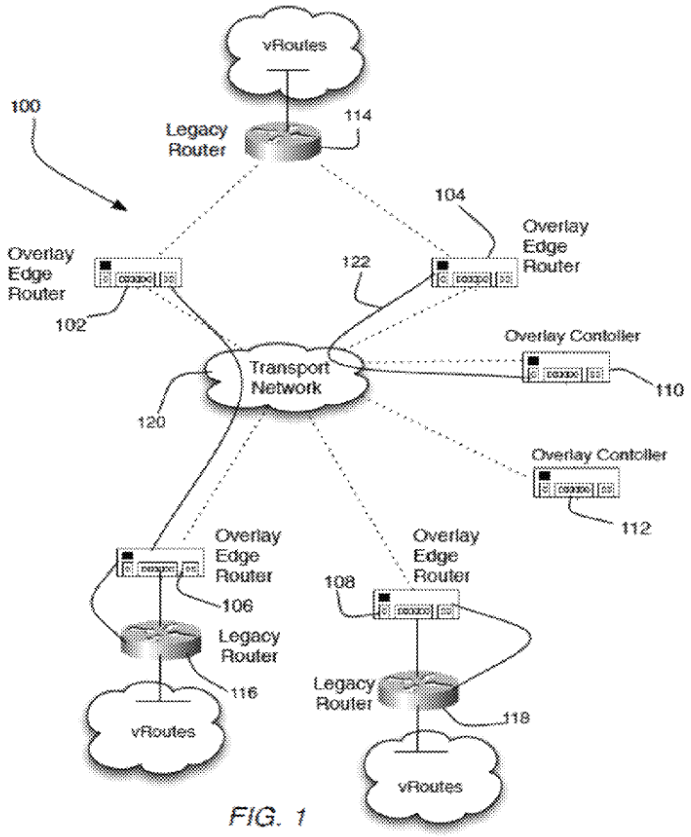
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| 1[preamble] | A method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network | The Reference discloses a method for use with a packet network including a network node for transporting packets between first and second entities under control of a controller that is external to the network node, the method comprising.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | node, the method comprising: | Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>OpenFlow at 6-7<br><br>  <br><br>Figure 1: Main components of an OpenFlow switch. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | **2 Switch Components**<br><br>An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The switch communicates with the controller and the controller manages the switch via the OpenFlow protocol.<br><br>Using the OpenFlow protocol, the controller can add, update, and delete *flow entries* in flow tables, both reactively (in response to packets) and proactively. Each flow table in the switch contains a set of flow entries; each flow entry consists of *match fields*, *counters*, and a set of *instructions* to apply to matching packets (see 5.2).<br><br>Matching starts at the first flow table and may continue to additional flow tables (see 5.1). Flow entries match packets in priority order, with the first matching entry in each table being used (see 5.3). If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on configuration of the table-miss flow entry: for example, the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table (see 5.4).<br><br>Instructions associated with each flow entry either contain actions or modify pipeline processing (see 5.9). Actions included in instructions describe packet forwarding, packet modification and group table<br><br>OpenFlow at 21<br><br>**6 OpenFlow Channel**<br><br>The OpenFlow channel is the interface that connects each OpenFlow switch to a controller. Through this interface, the controller configures and manages the switch, receives events from the switch, and sends packets out the switch.<br><br>Between the datapath and the OpenFlow channel, the interface is implementation-specific, however all OpenFlow channel messages must be formatted according to the OpenFlow protocol. The OpenFlow channel is usually encrypted using TLS, but may be run directly over TCP.<br><br>OpenFlow at 22 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | **6.1.1 Controller-to-Switch**<br><br>Controller/switch messages are initiated by the controller and may or may not require a response from the switch.<br><br>**Features**: The controller may request the capabilities of a switch by sending a features request; the switch must respond with a features reply that specifies the capabilities of the switch. This is commonly performed upon establishment of the OpenFlow channel.<br><br>**Configuration**: The controller is able to set and query configuration parameters in the switch. The switch only responds to a query from the controller.<br><br>**Modify-State**: Modify-State messages are sent by the controller to manage state on the switches. Their primary purpose is to add, delete and modify flow/group entries in the OpenFlow tables and to set switch port properties.<br><br>**Read-State**: Read-State messages are used by the controller to collect various information from from the switch, such as current configuration, statistics and capabilities.<br><br>**Packet-out**: These are used by the controller to send packets out of a specified port on the switch, and to forward packets received via Packet-in messages. Packet-out messages must contain a full packet or a buffer ID referencing a packet stored in the switch. The message must also contain a list of actions to be applied in the order they are specified; an empty action list drops the packet.<br><br>**Barrier**: Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations.<br><br>**Role-Request**: Role-Request messages are used by the controller to set the role of its OpenFlow channel, or query that role. This is mostly useful when the switch connects to multiple controllers (see 6.3.4).<br><br>**Asynchronous-Configuration**: The Asynchronous-Configuration message are used by the controller to set an additional filter on the asynchronous messages that it wants to receive on its OpenFlow channel, or to query that filter. This is mostly useful when the switch connects to multiple controllers (see 6.3.4) and commonly performed upon establishment of the OpenFlow channel.<br><br>**6.1.2 Asynchronous**<br><br>Asynchronous messages are sent without a controller soliciting them from a switch. Switches send asynchronous messages to controllers to denote a packet arrival, switch state change, or error. The four main asynchronous message types are described below.<br><br>**Packet-in:** Transfer the control of a packet to the controller. For all packets forwarded to the **CONTROLLER** reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers (see 5.12). Other processing, such as TTL checking, may also send packets to the controller using packet-in events. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of an overlay controller. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.")<br><br>Khan '478 at Figure 1 |

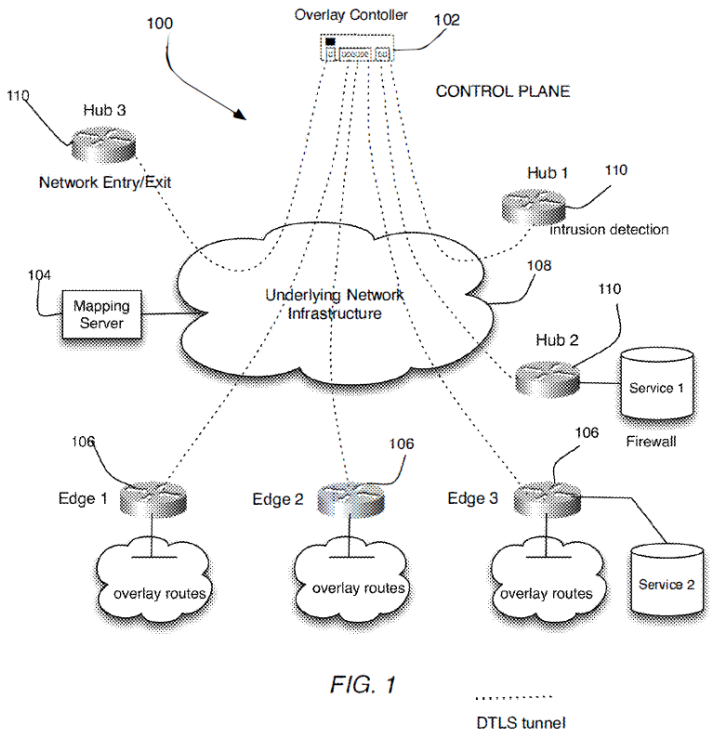
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by reference numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.")<br><br>Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing.<br>The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.")<br><br>Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI).<br>In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.")<br><br>Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the applicationflows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

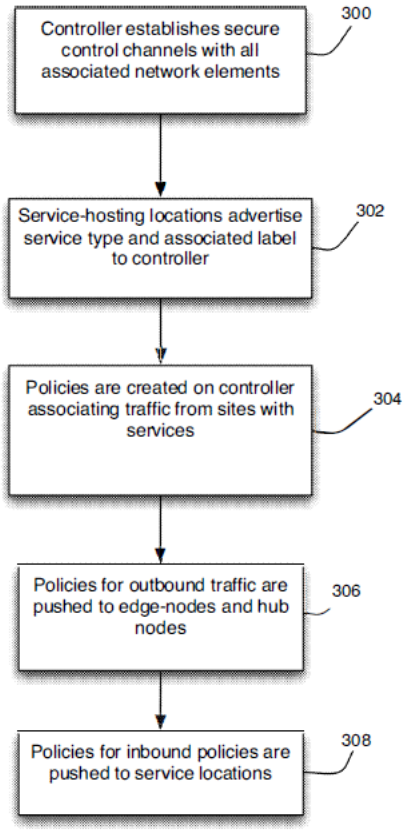



*FIG. 3*

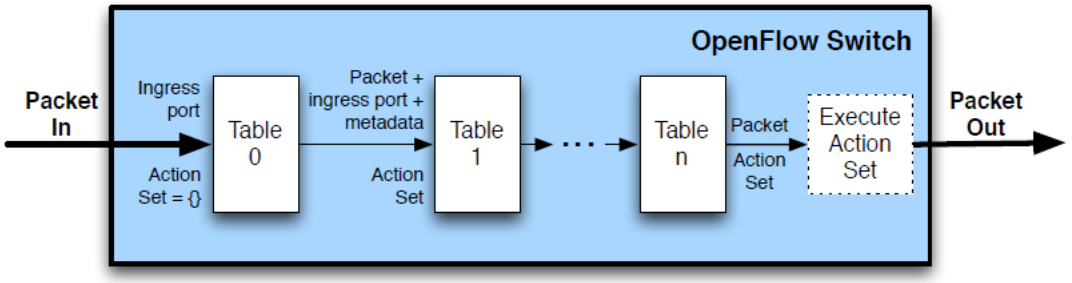
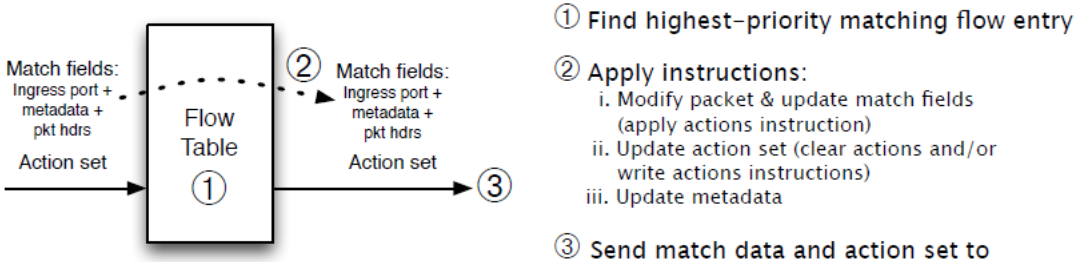
Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are related to the exact nature of the service chain that is being constructed. Two examples are provided below.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.) |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service-functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.")<br><br>Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |
| 1[a] | sending, by the controller to the network node over the packet network, an instruction and a packet-applicable criterion; | The Reference discloses sending, by the controller to the network node over the packet network, an instruction and a packet-applicable criterion.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Wang '735, Olofsson '254, and Kumar '739. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Below are examples of such references.<br><br>OpenFlow at 11<br><br>  <br><br>(a) Packets are matched against multiple tables in the pipeline<br><br>  <br><br>(b) Per-table packet processing<br><br>Figure 2: Packet flow through the processing pipeline<br><br>OpenFlow at 12 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

Table 1: Main components of a flow entry in a flow table.

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **priority**: matching precedence of the flow entry
- **counters**: to update for matching packets
- **instructions** to modify the action set or pipeline processing
- **timeouts**: maximum amount of time or idle time before flow is expired by the switch
- **cookie**: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion, not used when processing packets.

A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table. The flow entry that wildcards all fields (all fields omitted) and has priority equal 0 is called the table-miss flow entry (see 5.4).

## 5.3 Matching

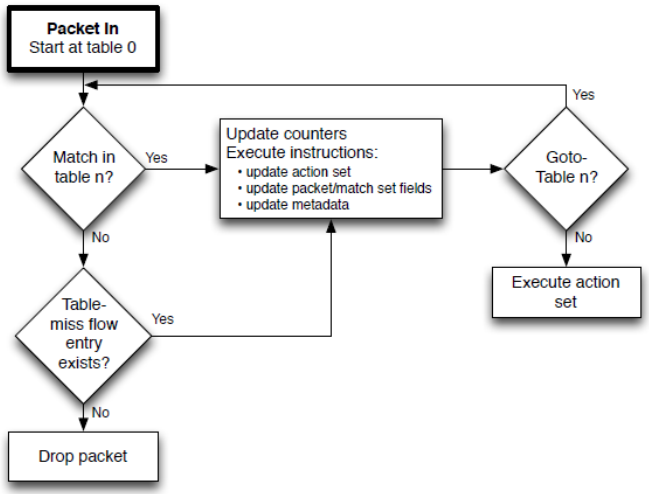
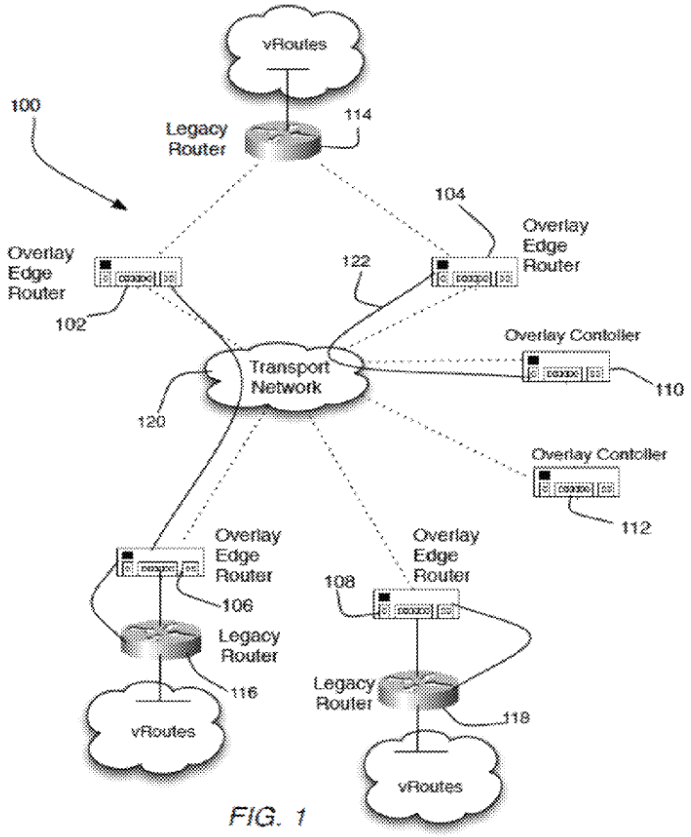



Figure 3: Flowchart detailing packet flow through an OpenFlow switch.

On receipt of a packet, an OpenFlow Switch performs the functions shown in Figure 3. The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables (see 5.1).

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | OpenFlow at 16<br>**5.9 Instructions**<br>Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.<br>A switch is not required to support all instruction types, just those marked *"Required Instruction"* below. The controller can also query the switch about which of the *"Optional Instruction"* it supports.<br>• *Optional Instruction:* **Meter** ***meter_id***: Direct packet to the specified meter. As the result of the metering, the packet may be dropped.<br>• *Optional Instruction:* **Apply-Actions** ***action(s)***: Applies the specific action(s) immediately, without any change to the Action Set. This instruction may be used to modify the packet between two tables or to execute multiple actions of the same type. The actions are specified as an action list (see 5.11).<br>• *Optional Instruction:* **Clear-Actions**: Clears all the actions in the action set immediately.<br>• *Required Instruction:* **Write-Actions** ***action(s)***: Merges the specified action(s) into the current action set (see 5.10). If an action of the given type exists in the current set, overwrite it, otherwise add it.<br>• *Optional Instruction:* **Write-Metadata** ***metadata / mask***: Writes the masked metadata value into the metadata field. The mask specifies which bits of the metadata register should be modified (i.e. new_metadata = old_metadata & ~mask \| value & mask).<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of an overlay controller. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.")<br><br>Khan '478 at Figure 1<br>  <br><br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by references |

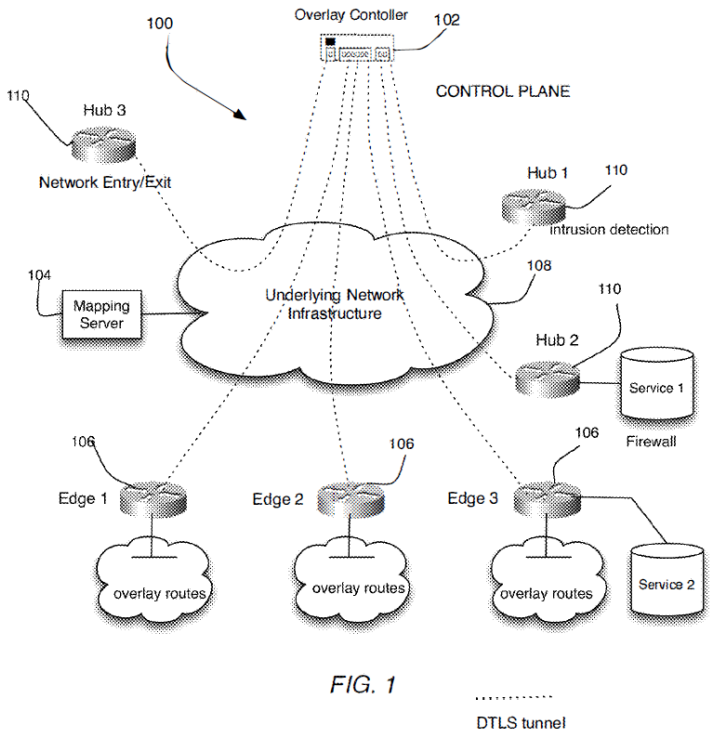
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.")<br><br>Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing.<br>The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI).<br>In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.")<br><br>Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Wang '735 at 4:30-45 ("The network device 12 may be, for example, a router ( e.g., ISR, ASR), integrated router/switch, or any other network device configured forrouting traffic. The router 12 may be an Internet-edge router in communication with an access switch or located at a branch office or data center, for example. The router 12 may be configured to enforce network policies, TCP throttling, provide network assessment/feedback, shape traf-fic (e.g., up/down speed, intelligent dropping), dynamically adjust queue bandwidth, or provide differentiated services, for example. The router 12 may also be operable to detect a performance issue, network status change (up or down), switch route, or perform preemption. In one or more embodi-ments, the router 12 is configured for Performance Routing (PfR), which is used to select a next hop to deliver the appli-cation traffic based on application performance measurement results and network resource status.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.")<br><br>Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the applicationflows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

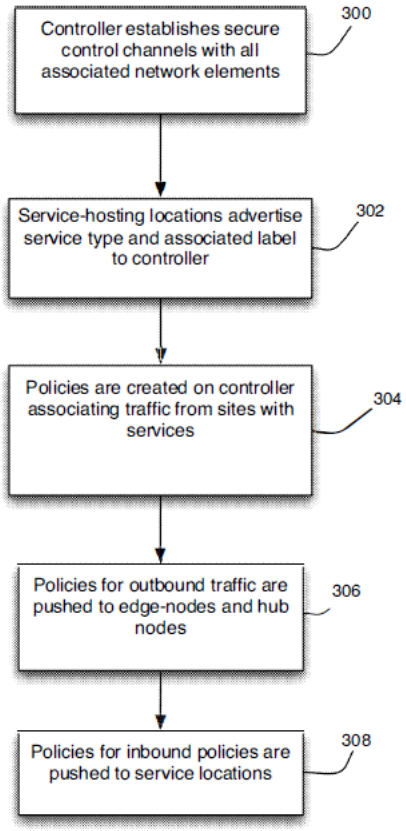



*FIG. 3*

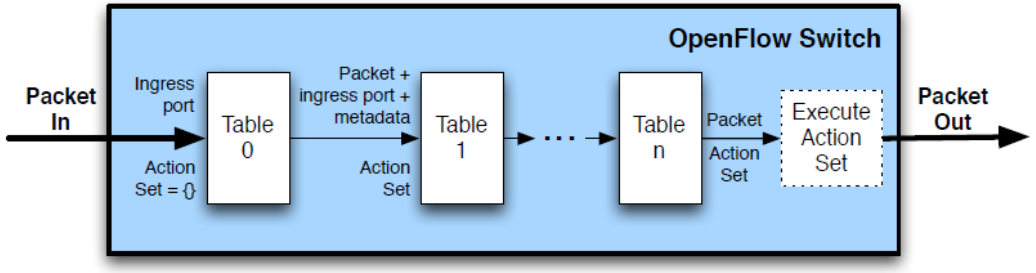
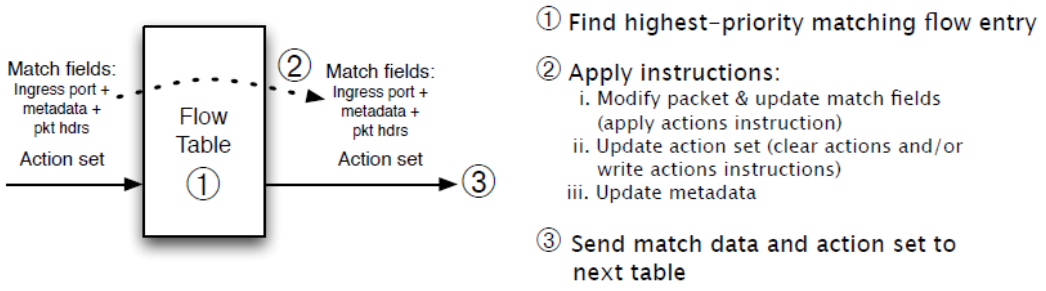
Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are related to the exact nature of the service chain that is being constructed. Two examples are provided below.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.) |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service-functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.")<br><br>Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |
| 1[b] | receiving, by the network node from the controller, the instruction and the criterion; | The Reference discloses receiving, by the network node from the controller, the instruction and the criterion.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | OpenFlow at 11   (a) Packets are matched against multiple tables in the pipeline   (b) Per-table packet processing Figure 2: Packet flow through the processing pipeline OpenFlow at 12 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

Table 1: Main components of a flow entry in a flow table.

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **priority**: matching precedence of the flow entry
- **counters**: to update for matching packets
- **instructions** to modify the action set or pipeline processing
- **timeouts**: maximum amount of time or idle time before flow is expired by the switch
- **cookie**: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion, not used when processing packets.

A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table. The flow entry that wildcards all fields (all fields omitted) and has priority equal 0 is called the table-miss flow entry (see 5.4).

## 5.3 Matching

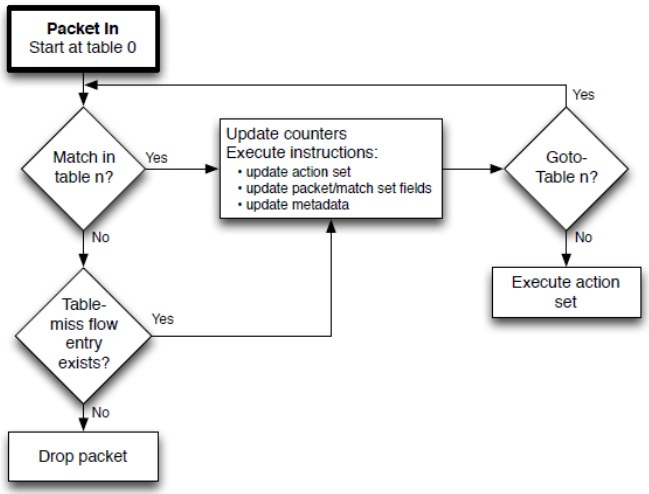
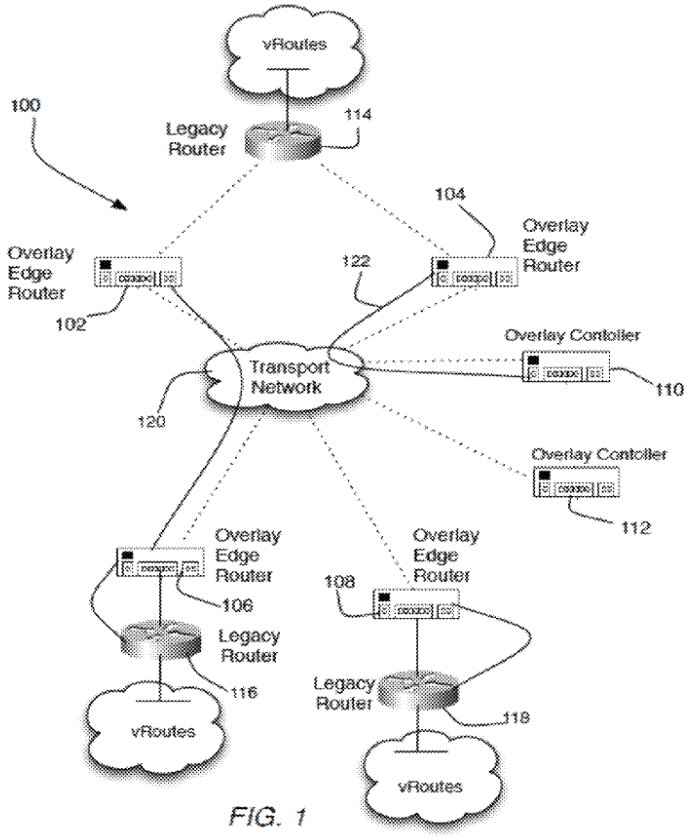



Figure 3: Flowchart detailing packet flow through an OpenFlow switch.

On receipt of a packet, an OpenFlow Switch performs the functions shown in Figure 3. The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables (see 5.1).

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | OpenFlow at 16<br>**5.9 Instructions**<br>Each flow entry contains a set of instructions that are executed when a packet matches the entry. These instructions result in changes to the packet, action set and/or pipeline processing.<br>A switch is not required to support all instruction types, just those marked *"Required Instruction"* below. The controller can also query the switch about which of the *"Optional Instruction"* it supports.<br>• *Optional Instruction:* **Meter** ***meter_id***: Direct packet to the specified meter. As the result of the metering, the packet may be dropped.<br>• *Optional Instruction:* **Apply-Actions** ***action(s)***: Applies the specific action(s) immediately, without any change to the Action Set. This instruction may be used to modify the packet between two tables or to execute multiple actions of the same type. The actions are specified as an action list (see 5.11).<br>• *Optional Instruction:* **Clear-Actions**: Clears all the actions in the action set immediately.<br>• *Required Instruction:* **Write-Actions** ***action(s)***: Merges the specified action(s) into the current action set (see 5.10). If an action of the given type exists in the current set, overwrite it, otherwise add it.<br>• *Optional Instruction:* **Write-Metadata** ***metadata / mask***: Writes the masked metadata value into the metadata field. The mask specifies which bits of the metadata register should be modified (i.e. new_metadata = old_metadata & ~mask \| value & mask).<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of an overlay controller. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.")<br><br>Khan '478 at Figure 1<br>  <br><br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by references |

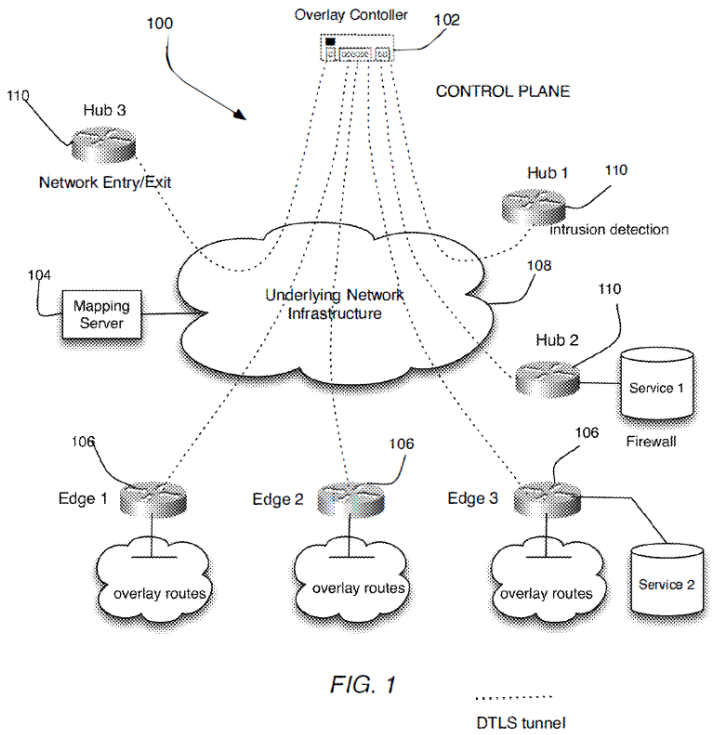
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.")<br><br>Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing.<br>The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI).<br>In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.")<br><br>Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Wang '735 at 4:30-45 ("The network device 12 may be, for example, a router ( e.g., ISR, ASR), integrated router/switch, or any other network device configured forrouting traffic. The router 12 may be an Internet-edge router in communication with an access switch or located at a branch office or data center, for example. The router 12 may be configured to enforce network policies, TCP throttling, provide network assessment/feedback, shape traf-fic (e.g., up/down speed, intelligent dropping), dynamically adjust queue bandwidth, or provide differentiated services, for example. The router 12 may also be operable to detect a performance issue, network status change (up or down), switch route, or perform preemption. In one or more embodi-ments, the router 12 is configured for Performance Routing (PfR), which is used to select a next hop to deliver the appli-cation traffic based on application performance measurement results and network resource status.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.")<br><br>Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the applicationflows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

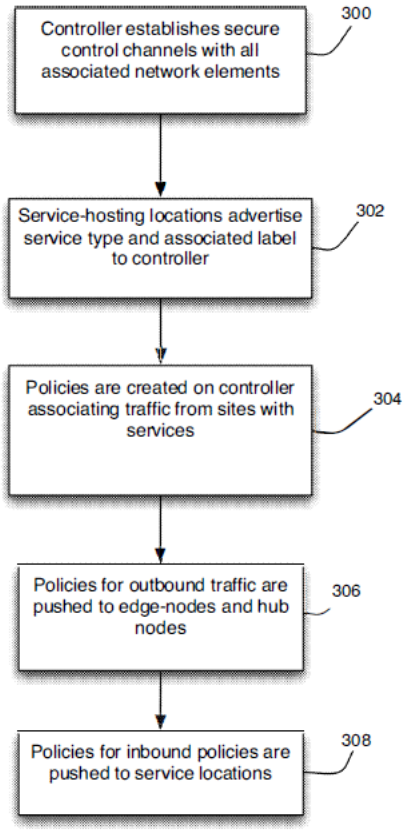



*FIG. 3*

Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated

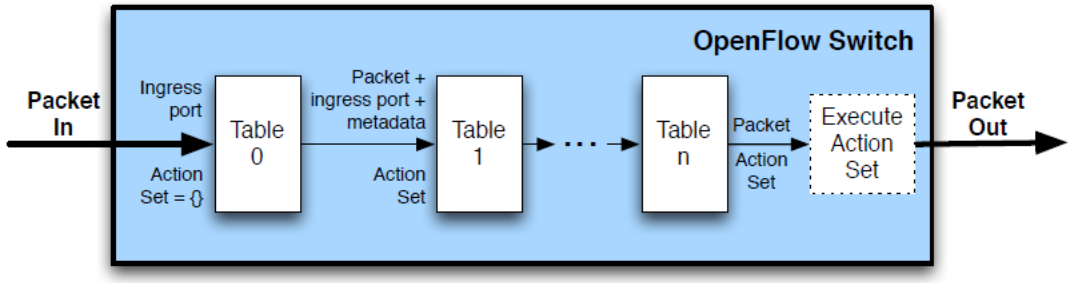
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are related to the exact nature of the service chain that is being constructed. Two examples are provided below.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.) |

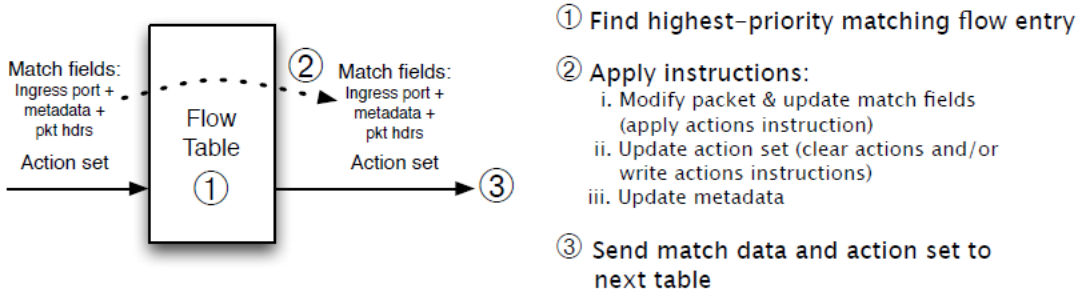
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service-functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.")<br><br>Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |
| 1[c] | receiving, by the network node from the first entity over the packet network, a packet addressed to the second entity; | The Reference discloses receiving, by the network node from the controller, the instruction and the criterion.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | OpenFlow at 11 |




(a) Packets are matched against multiple tables in the pipeline




(b) Per-table packet processing

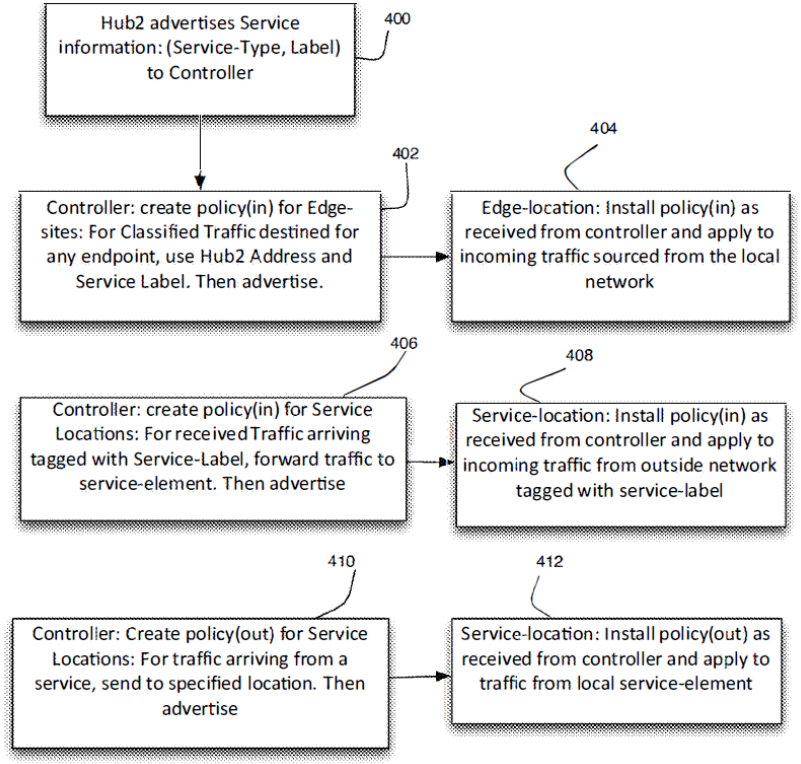
Figure 2: Packet flow through the processing pipeline

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | The flow tables of an OpenFlow switch are sequentially numbered, starting at 0. Pipeline processing always starts at the first flow table: the packet is first matched against flow entries of flow table 0. Other flow tables may be used depending on the outcome of the match in the first table.<br><br>When processed by a flow table, the packet is matched against the flow entries of the flow table to select a flow entry (see 5.3). If a flow entry is found, the instruction set included in that flow entry is executed, those instructions may explicitly direct the packet to another flow table (using the Goto Instruction, see 5.9), where the same process is repeated again. A flow entry can only direct a packet to a flow table number which is greater than its own flow table number, in other words pipeline processing can only go forward and not backward. Obviously, the flow entries of the last table of the pipeline can not include the Goto instruction. If the matching flow entry does not direct packets to another flow table, pipeline processing stops at this table. When pipeline processing stops, the packet is processed with its associated action set and usually forwarded (see 5.10).<br><br>If a packet does not match a flow entry in a flow table, this is a table miss. The behavior on a table miss depends on the table configuration (see 5.4). A table-miss flow entry in the flow table may specify how to process unmatched packets: Options include dropping them, passing them to another table or sending them to the controller over the control channel via packet-in messages (see 6.1.2).<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at 3:58-67 (" In one embodiment, secure tunnels may be established between one overlay edge router (OER) and another overlay edge router (OER). For example, reference numeral 124 shows a secure tunnel that may exist as an IPSec tunnel between the overlay edge router (OER) 102 and the overlay edge router (OER) 106. The tunnel 124 is through the transport network 120 and is used to transport data between its end points in a secure manner. The plurality of tunnels established between the various overlay edge routers (OERs) together form a secure overlay data plane (ODP).") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Khan '478 at 4:18-26 (" In one embodiment, an overlay route may include the prefixes that establish reachability between endpoints. An overlay route may represent services in a central data center, services at a branch office or collections of hosts and other endpoints in any location of the overlay network. An overlay route may require and resolve onto TLOCs for functional forwarding. In comparison with BGP, an overlay route may be considered to be the equivalent of a prefix carried in any of the BGP AFI/SAFI constructs.")<br><br>Khan '478 at 4:27-37 (" In one embodiment, a transport locator (TLOC) ties an overlay route to a physical location. The TLOC is the only visible entity of the OMP routing domain to the underlying transport network 120, and is reachable via routing in the transport network 120. A TLOC can be directly reachable via an entry in the routing table of the physical network or be represented by a prefix residing on the outside of a NAT device, also present in the aforementioned routing table. The TLOC acts as the next-hop for overlay routes, to continue the BGP-analogy.")<br><br>Khan '478 at 5:53-61 (" TLOCs (Transport locations) are the location ids, e.g. a WAN interface connecting into a carrier. TLOCs are denoted by {System-IP, Link-color} as described below. The reason for not using an interface IP address to denote a TLOC is that IP addresses can move or change (e.g. if it is DHCP assigned). Using {system-IP, color} to denote TLOCs ensures that a transport endpoint can be identified irrespec-tive of the interface IP addressing.")<br><br>Wang '735 at 3:41-64 ("The embodiments described herein operate in the context of a data communication system including multiple network elements. Referring now to the drawings, and first to FIG. 1, an example of a network in which embodiments described herein may be implemented is shown. The communication system comprises a plurality of endpoints 10 in communica-tion through a plurality of network devices ( e.g., routers) 12 and over networks 14. The communication system may include any number of networks ( e.g., local area network, metropolitan area network, wide area network, enterprise network, Internet, intranet, radio access network, public switched network, or any other network or combination of networks). The flow path between the endpoints 10 may include any number or type of intermediate nodes ( e.g., rout-ers, switches, gateways, management stations, appliances, or other network devices), which facilitate passage of data between the |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | endpoints. Also, there may be any number of endpoints 10. The endpoints 10 may be located at a branch office, for example, and in communication with an ISR (Inte-grated Services Router) 12 connected to a WAN access link 13. The ISRs may be in communication with ASRs (Aggre-gated Services Router) 12 operating at the network edge, for example. The routers 12 communicate over a wide area net-work.")<br><br>Wang '735 at 4:9-29 ("The endpoints 10 are configured to originate or terminate communications over the network. The endpoints 10 may be any device or combination of devices configured for receiv-ing, transmitting, or receiving and transmitting traffic. As described below, the network device 12 receives application traffic from one or more endpoints. This includes, for example, receiving traffic from one or more upstream net- work devices. Traffic may include audio, video, text, or other data or combination thereof. The endpoint 10 may be, for example, a server that stores media locally or receives the media from another server or media source via another net-work, satellite, cable, or any other communication device. The endpoint 10 may also be, for example, a personal com-puter, set-top box, personal digital assistant (PDA), VoIP phone, tablet, Internet connected television, cellular tele-phone, TelePresence device, media center device, or any other network device that receives or transmits packets. As described below, the endpoints 10 may be configured for FEC (Forward Error Correction), rate-adaptation, error conceal-ment, RTCP (Real-time Transport Control Protocol) feed-back or other protocols or technologies.")<br><br>Wang '735 at 4:30-45 ("The network device 12 may be, for example, a router ( e.g., ISR, ASR), integrated router/switch, or any other network device configured forrouting traffic. The router 12 may be an Internet-edge router in communication with an access switch or located at a branch office or data center, for example. The router 12 may be configured to enforce network policies, TCP throttling, provide network assessment/feedback, shape traf-fic (e.g., up/down speed, intelligent dropping), dynamically adjust queue bandwidth, or provide differentiated services, for example. The router 12 may also be operable to detect a performance issue, network status change (up or down), switch route, or perform preemption. In one or more embodi-ments, the router 12 is configured for Performance |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Routing (PfR), which is used to select a next hop to deliver the appli-cation traffic based on application performance measurement results and network resource status.")<br><br>Wang '735 at 5:7-19 ("Nodes 12 configured with the optimization system 18 are preferably operable to automatically discover other nodes configured with the optimization system on the media path. Notifications and requests may be sent among these discovered optimization system devices 12 to effectively manage the media traffic bandwidth and to optimize the media appli-cation performance. The received notifications or requests from adjacent optimization system devices (upstream and downstream devices along the media path) can generate additional requests or notifications to other components ( e.g., service routing, optimizer, policy manager, scheduler). The nodes 12 may use RSVP (Resource Reservation Protocol) for communication with one another, for example.")<br><br>Olofsson '254 at Figure 4 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>*FIG. 4*<br>Olofsson '254 at Figure 5 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

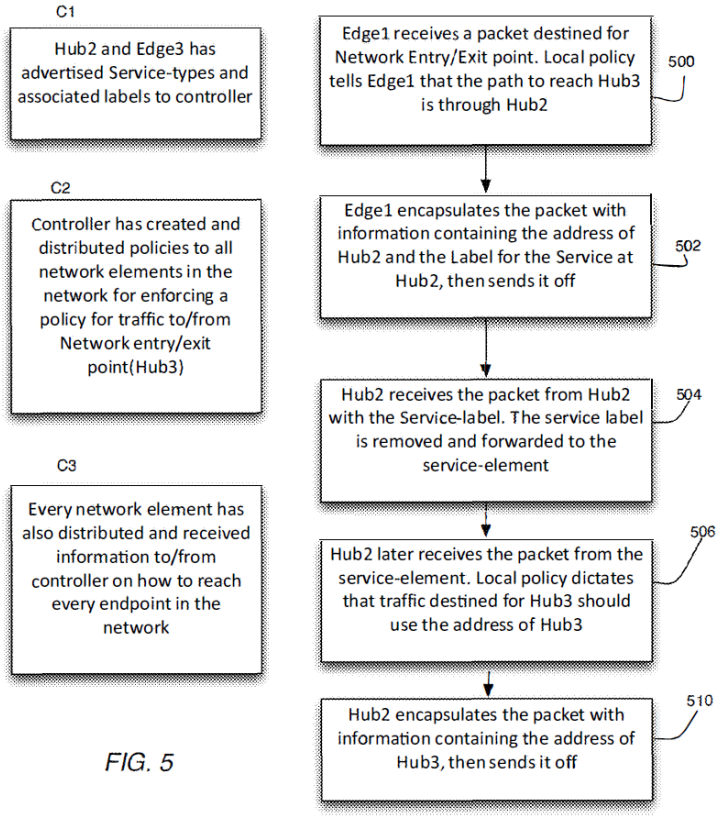



*FIG. 5*

Olofsson '254 at 4:60-5:28 ("The setup and distribution of policies will take place as follows:

Block 400: Hub2 advertises Service information: (Ser-vice-Type, Label) to the controller 102.

Block 402: The overlay controller 102 constructs the set of required policies and distributes them accordingly. This means that each node involved will be assigned a policy for managing the required traffic flow.

Block 404: Every Edge-router receives an outbound policy (for traffic towards the Internet) stipulating that all the traffic matching the routes received from Hub3, will be encapsulated

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | in a packet with a Service-label of 1, matching the Firewall Service, and a next-hop address of Hub2. This will ensure all traffic destined for the Internet is using the tunnel from the Edge-router to Hub2<br><br>Block 406: The controller 102 creates a policy (in) for Service Locations: For received Traffic arriving tagged with Service-Label, forward traffic to service-element and adver-tises the policy thus created to each hub.<br><br>Block 408: Hub2 receives an inbound policy (for traffic received on its external interface), stipulating that all received traffic matching Service Label 1 is sent to the firewall. Exactly how the traffic is forwarded is a local decision.<br><br>Block 410: Hub2 receives an outbound policy (for traffic received from the Firewall) stipulating that the traffic des-tined for the Internet must be encapsulated with a next-hop address ofHub3, taking the traffic to the Internet. Any traffic destined for an Edge-router in the overlay network is sent with an encapsulation of that Edge-routers next-hop address<br><br>Block 412: Hub3 receives an inbound policy (for traffic received from the Internet) stipulating that all traffic matching a destination advertised from an Edge-router is encap-sulated in a packet with a Service-label of 1, matching the Firewall Service, and a next-hop address of Hub2")<br><br>Olofsson '254 at 5:33-67 ("Using the same setup as in the prior example, but adding that all traffic returning from the Internet must not only pass through the Firewall Service, but also pass through an Intrusion Detection Service, the modification required to the infrastructure is the following:<br><br>Hubl hosts the Intrusion Detection service and advertises a service with a Label of 2 and a Service-type of Intrusion Detection.<br><br>The changes in setup and distribution of policies is following:<br><br>Hub2, hosting the firewall service, has one addition to its outbound policy. All traffic destined for any Edge-router in the overlay network must be encapsulated with a Service-label of 2 and a next-hop address of Hubl. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Hubl is equipped with policies identical to what Hub2 had in the previous example, with the differences being that the label matching is done on Service-label 2 for inbound traffic. FIG. 5 shows a flowchart corresponding to this example of service chaining. Referring the FIG. 5, Cl to C3 are conditions and the processing blocks are as provided below:<br><br>Block 500: Edge1 receives a packet destined for Network Entry/Exit point. Local policy tells Edge! that the path to reach Hub3 is through Hub2<br><br>Block 502: Edge! encapsulates the packet with informa-tion containing the address of Hub2 and the Label for the Service at Hub2, then sends it off.<br><br>Block 504: Hub2 receives the packet from Hub2 with the Service-label. The service label is removed and forwarded to the service-element.<br><br>Block 506 Hub2 later receives the packet from the ser-vice-element. Local policy dictates that traffic destined for Hub3 should use the address of Hub3.<br><br>Block 508 Hub2 encapsulates the packet with information containing the address of Hub3, then sends it off.")<br><br>Kumar '739 at 5:34-51 ("In the example of FIG. 2, classification and mapping logic 75 selects service-functions from several different service nodes. In particular, classification and mapping logic 75 selects service-functions f1 and f2 at service node 35, ser-vice-functions f6 and f7 at service node 40, and service-function f10 at service node 45. The path for service-function chain SFC1 selected by classification and mapping logic 75 is shown in FIG. 2 by broken line 100. The classifier 30 sends traffic 90 along the path 100 using one or more L2/L3/L4 service overlays in the network. In other words, a service header is appended to the traffic 90 for forwarding through the service chain and the service header enables the carrying of service metadata in addition to the original data/payload.<br><br>Service-function f10 is the end of the service-function chain SFC1. After processing the traffic 90, the service-function f10 may forward the traffic 90 to its original or other destination.") |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Kumar '739 at 7:11-25 ("The path for service-function sub-chain SFC3a selected by classification and mapping logic 75 is shown in FIG. 3 by broken line 160, while the path for service-function sub-chain SFC3b selected by classification and mapping logic 75 is shown in FIG. 3 by broken line 170. The classifier 30 sends traffic 150 along the path 160 using one or more L2/L3/L4 service overlays in the network. Once the traffic 150 is serviced by f7, the traffic 150 is re-classified at classifier 130 to choose SFC3b and is steered through f10.<br><br>After processing the traffic 150, the service-function f10 may forward the traffic 150 to its original or other destination. In the above example, the first primary classifier 30 only performs mapping for the first sub-chain SFC3a while the second primary classifier 130 performs mapping for the second sub-chain SFC3b.") |
| 1[d] | checking, by the network node, if the packet satisfies the criterion; | The Reference discloses checking, by the network node, if the packet satisfies the criterion.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, and Balakrishnan.<br><br>Below are examples of such references.<br><br>OpenFlow at 6-7 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

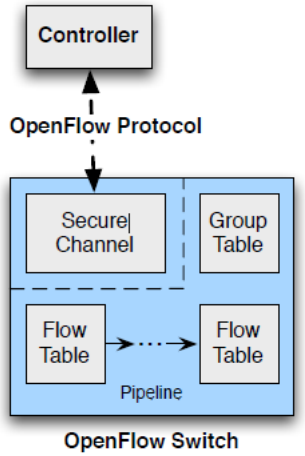



Figure 1: Main components of an OpenFlow switch.

## 2 Switch Components

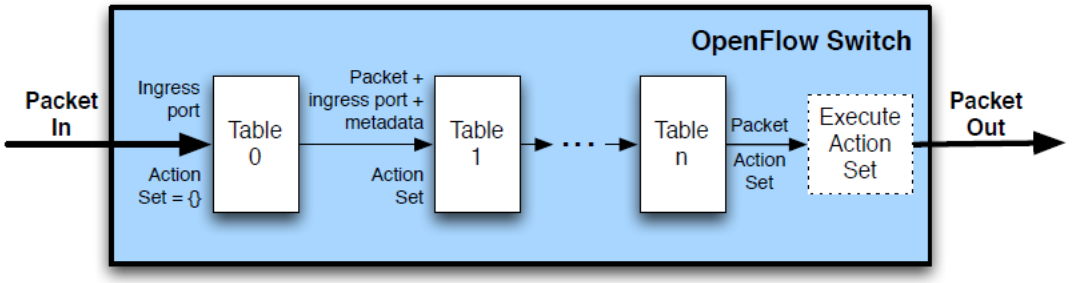
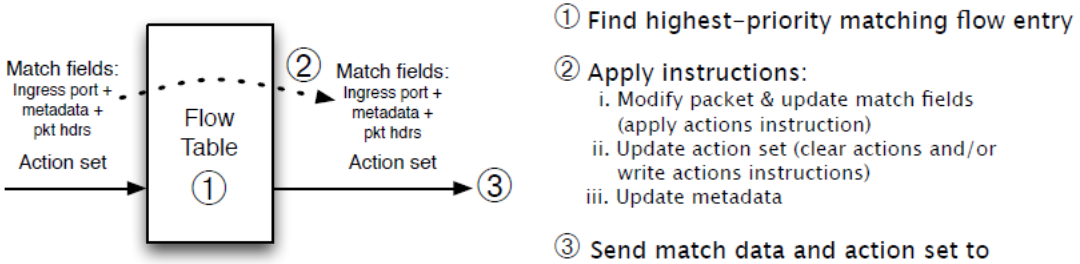
An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The switch communicates with the controller and the controller manages the switch via the OpenFlow protocol.

Using the OpenFlow protocol, the controller can add, update, and delete *flow entries* in flow tables, both reactively (in response to packets) and proactively. Each flow table in the switch contains a set of flow entries; each flow entry consists of *match fields*, *counters*, and a set of *instructions* to apply to matching packets (see 5.2).

Matching starts at the first flow table and may continue to additional flow tables (see 5.1). Flow entries match packets in priority order, with the first matching entry in each table being used (see 5.3). If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on configuration of the table-miss flow entry: for example, the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table (see 5.4).

Instructions associated with each flow entry either contain actions or modify pipeline processing (see 5.9). Actions included in instructions describe packet forwarding, packet modification and group table

OpenFlow at 11

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>(a) Packets are matched against multiple tables in the pipeline<br>  <br>(b) Per-table packet processing<br>Figure 2: Packet flow through the processing pipeline |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | The flow tables of an OpenFlow switch are sequentially numbered, starting at 0. Pipeline processing always starts at the first flow table: the packet is first matched against flow entries of flow table 0. Other flow tables may be used depending on the outcome of the match in the first table.<br><br>When processed by a flow table, the packet is matched against the flow entries of the flow table to select a flow entry (see 5.3). If a flow entry is found, the instruction set included in that flow entry is executed, those instructions may explicitly direct the packet to another flow table (using the Goto Instruction, see 5.9), where the same process is repeated again. A flow entry can only direct a packet to a flow table number which is greater than its own flow table number, in other words pipeline processing can only go forward and not backward. Obviously, the flow entries of the last table of the pipeline can not include the Goto instruction. If the matching flow entry does not direct packets to another flow table, pipeline processing stops at this table. When pipeline processing stops, the packet is processed with its associated action set and usually forwarded (see 5.10).<br><br>If a packet does not match a flow entry in a flow table, this is a table miss. The behavior on a table miss depends on the table configuration (see 5.4). A table-miss flow entry in the flow table may specify how to process unmatched packets: Options include dropping them, passing them to another table or sending them to the controller over the control channel via packet-in messages (see 6.1.2).<br><br>OpenFlow at 12 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

Table 1: Main components of a flow entry in a flow table.

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **priority**: matching precedence of the flow entry
- **counters**: to update for matching packets
- **instructions** to modify the action set or pipeline processing
- **timeouts**: maximum amount of time or idle time before flow is expired by the switch
- **cookie**: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion, not used when processing packets.

A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table. The flow entry that wildcards all fields (all fields omitted) and has priority equal 0 is called the table-miss flow entry (see 5.4).

## 5.3 Matching

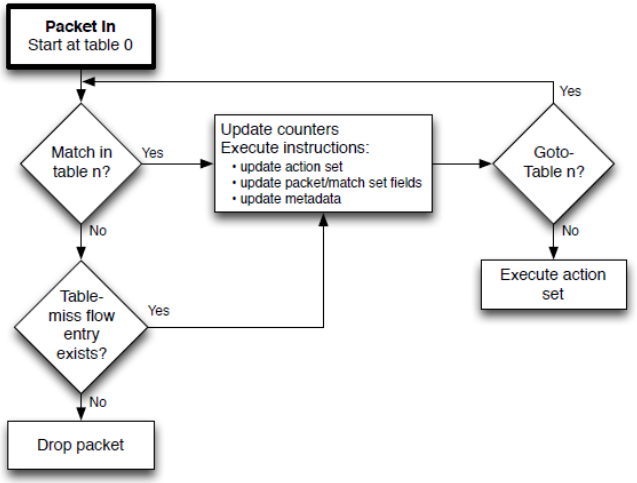
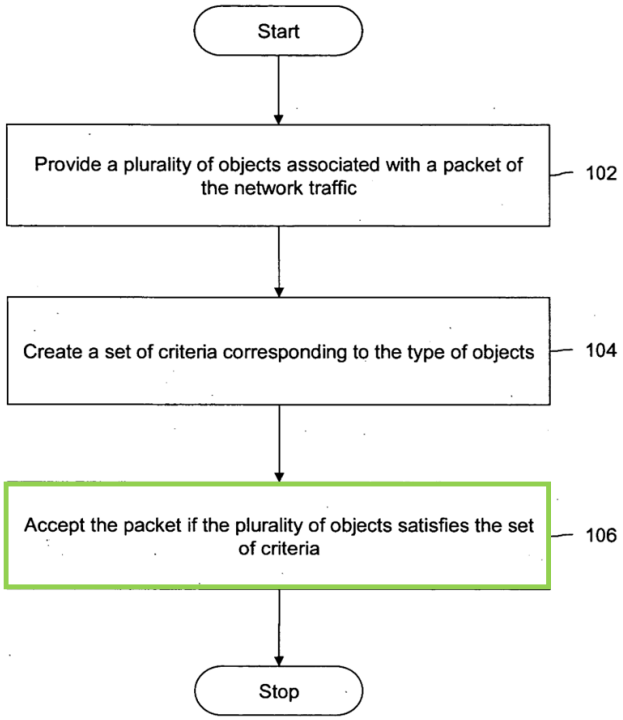
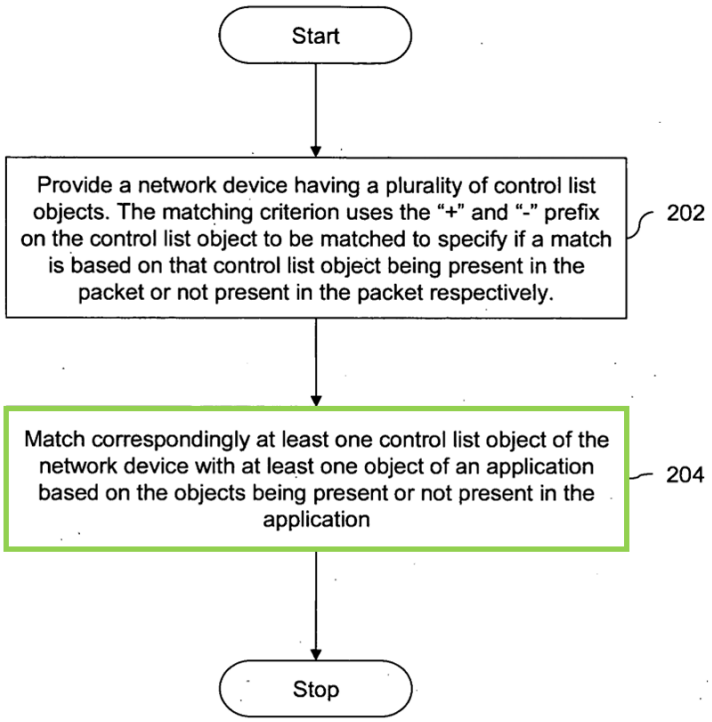


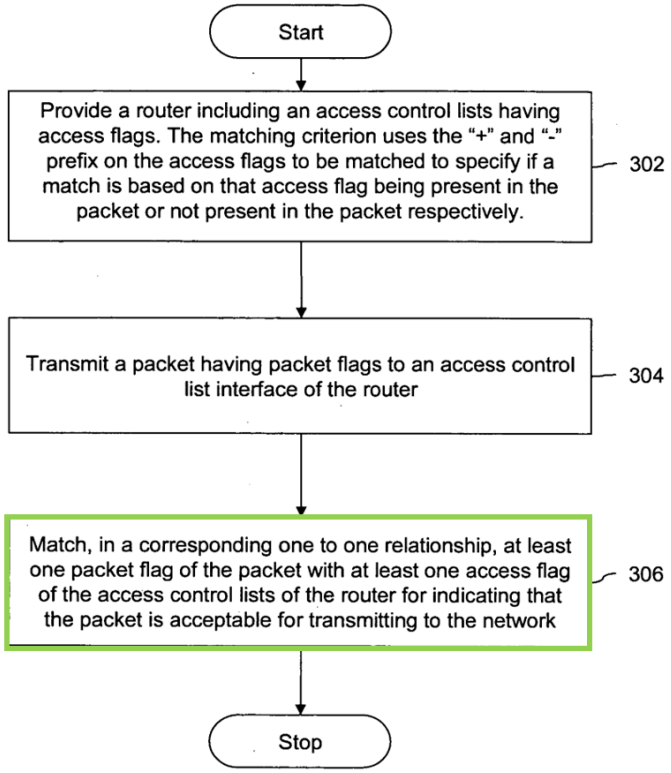

Figure 3: Flowchart detailing packet flow through an OpenFlow switch.

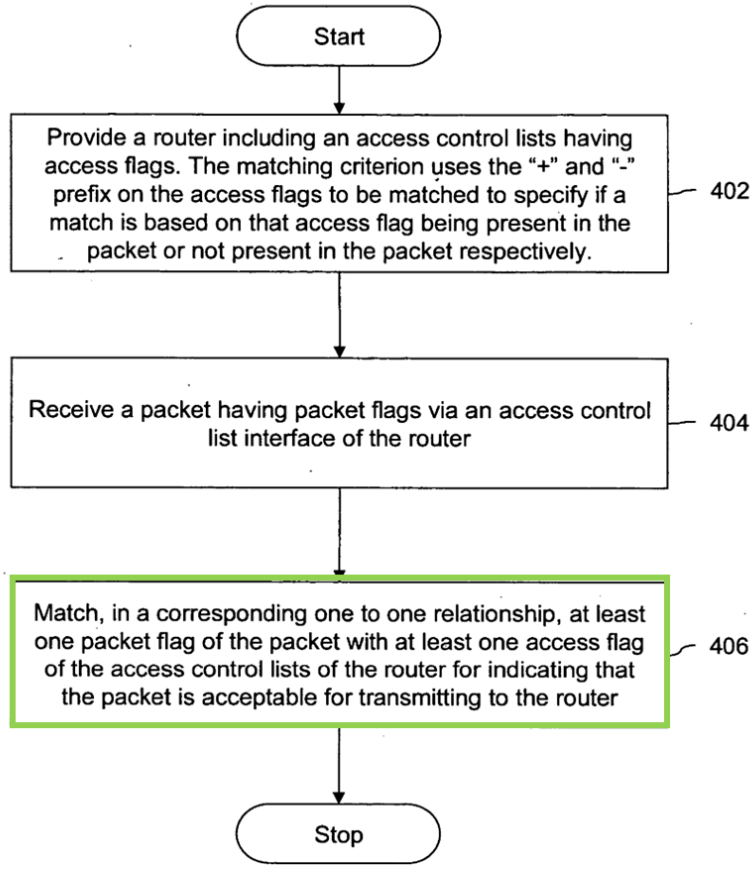
On receipt of a packet, an OpenFlow Switch performs the functions shown in Figure 3. The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables (see 5.1).

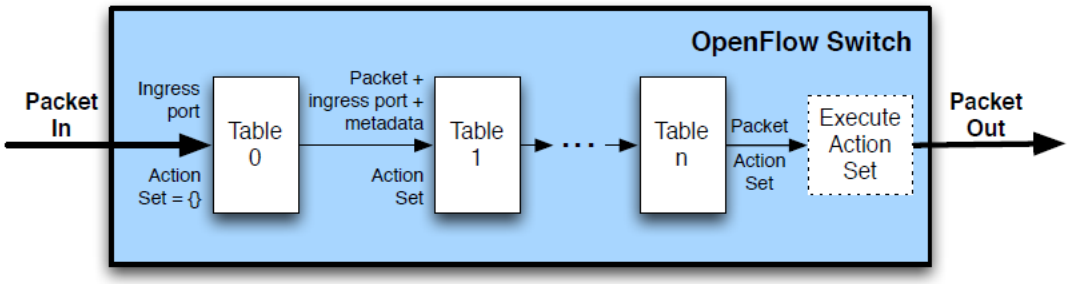
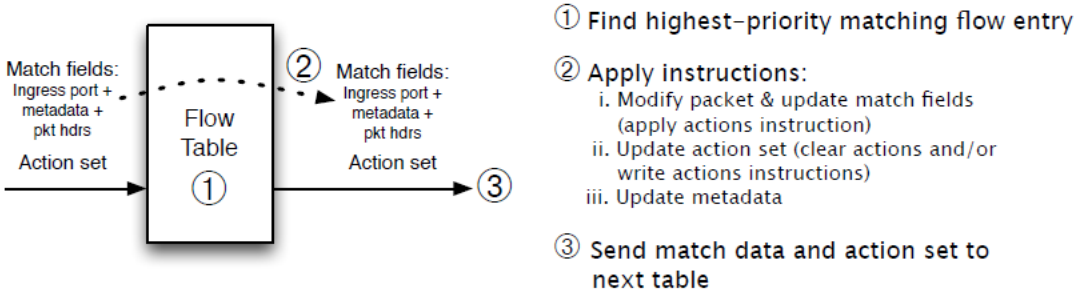
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Balakrishnan at [0010] ("In an embodiment of the invention, a system for managing network traffic in a network device is provided. The system includes means for providing a plurality of objects associated with a packet of the network traffic; and means for correspondingly matching at least one object of the network device with at least one object associated with the packet.").<br><br>Balakrishnan at [0021] ("At step 106, the packet is accepted, if the plurality of objects satisfies the set of criteria. For example, the set of criteria may include a 'set criterion' such that the packets having the TCP flags syn and ack set, are accepted. Further, the set of criteria may include a 'not-set criterion', for example, the packets having TCP flag fin not set are accepted.").<br><br>Balakrishnan at [0022] ("At step 202, a network device having a plurality of control list objects is provided. The control list objects may be, for example, filtering TCP packets in an Access Control List (ACL). The ACL is generally implemented as a data structure such as a tree that has the information (to be used by the network operating system) to determine the access rights of each packet. At step 204, at least one control list object of the network device is correspondingly matched with at least one object of the application. The matching is performed based on the objects being present or not present in the application. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG. 1. In an embodiment of the invention, a '+' sign is prefixed to a control list object in the 'set criterion', to signify that the packet is considered a match if the control list object is present in the packet. In another embodiment of the invention, a '−' sign is prefixed to a control list object, to signify that the packet is considered a match if the control list object is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device.").<br><br>Balakrishnan at [0023] ("At step **302**, a router including an ACL having match condition based on packet flags is provided. At step **304**, a packet having packet flags is transmitted to an interface of the router that has an ACL. Subsequently, at least one packet flag of the packet is matched with at least one packet flag as specified in the ACL of the router, at step **306**. The matching is performed based on the matching criteria for indicating that the packet is acceptable for transmitting to the network. If the packet flag(s) match the |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | conditions in the ACL, then the packet is transmitted to the network. If the packet flag(s) do not satisfy the conditions in the ACL, the packet is not transmitted.").<br>  <br><br>FIG. 1<br>Balakrishnan Figure 1 (annotation added). |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 2<br>Balakrishnan Figure 2 (annotation added). |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 3<br>Balakrishnan Figure 3 (annotation added). |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | |  <br>FIG. 4<br>Balakrishnan Figure 4 (annotation added). |
| 1[e] | responsive to the packet not satisfying | The Reference discloses responsive to the packet not satisfying the criterion, sending, by the network node over the packet network, the packet to the second entity. |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | the criterion, sending, by the network node over the packet network, the packet to the second entity; and | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 11<br><br>  <br><br>(a) Packets are matched against multiple tables in the pipeline<br><br>  <br><br>(b) Per-table packet processing<br><br>Figure 2: Packet flow through the processing pipeline |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | The flow tables of an OpenFlow switch are sequentially numbered, starting at 0. Pipeline processing always starts at the first flow table: the packet is first matched against flow entries of flow table 0. Other flow tables may be used depending on the outcome of the match in the first table.<br><br>When processed by a flow table, the packet is matched against the flow entries of the flow table to select a flow entry (see 5.3). If a flow entry is found, the instruction set included in that flow entry is executed, those instructions may explicitly direct the packet to another flow table (using the Goto Instruction, see 5.9), where the same process is repeated again. A flow entry can only direct a packet to a flow table number which is greater than its own flow table number, in other words pipeline processing can only go forward and not backward. Obviously, the flow entries of the last table of the pipeline can not include the Goto instruction. If the matching flow entry does not direct packets to another flow table, pipeline processing stops at this table. When pipeline processing stops, the packet is processed with its associated action set and usually forwarded (see 5.10).<br><br>If a packet does not match a flow entry in a flow table, this is a table miss. The behavior on a table miss depends on the table configuration (see 5.4). A table-miss flow entry in the flow table may specify how to process unmatched packets: Options include dropping them, passing them to another table or sending them to the controller over the control channel via packet-in messages (see 6.1.2).<br><br>OpenFlow at 12 |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

Table 1: Main components of a flow entry in a flow table.

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **priority**: matching precedence of the flow entry
- **counters**: to update for matching packets
- **instructions** to modify the action set or pipeline processing
- **timeouts**: maximum amount of time or idle time before flow is expired by the switch
- **cookie**: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion, not used when processing packets.

A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table. The flow entry that wildcards all fields (all fields omitted) and has priority equal 0 is called the table-miss flow entry (see 5.4).

## 5.3 Matching

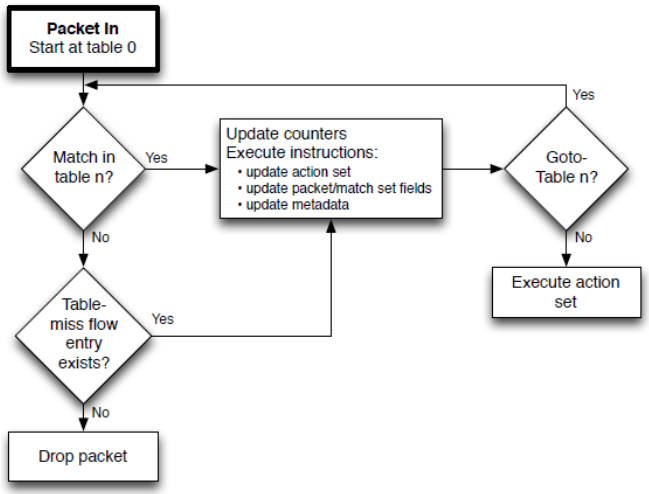



Figure 3: Flowchart detailing packet flow through an OpenFlow switch.

On receipt of a packet, an OpenFlow Switch performs the functions shown in Figure 3. The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables (see 5.1).

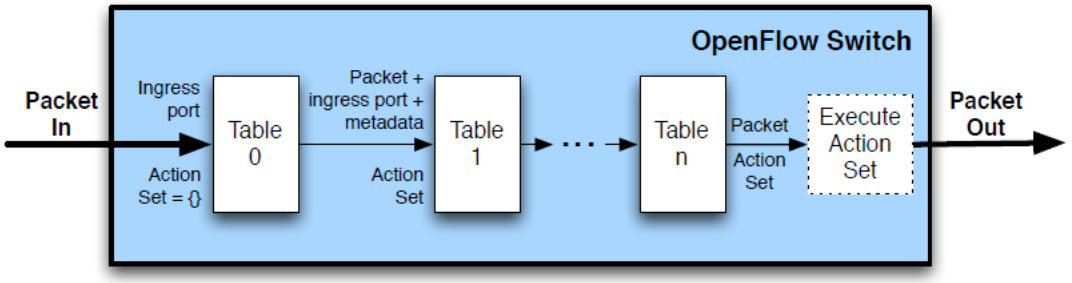
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| 1[f] | responsive to the packet satisfying the criterion, sending the packet, by the network node over the packet network, to an entity that is included in the instruction and is other than the second entity. | The Reference discloses responsive to the packet satisfying the criterion, sending the packet, by the network node over the packet network, to an entity that is included in the instruction and is other than the second entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and Balakrishnan.<br><br>Below are examples of such references.<br><br>Balakrishnan at [0020] ("In an embodiment of the invention, a '+' sign is prefixed to a packet flag in the 'set criterion', to signify that the packet is considered a match if the packet flag is present in the packet. In another embodiment of the invention, a '−' sign is prefixed to a packet flag, to signify that the packet is considered a match if the packet flag is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device.").<br><br>Balakrishnan at [0021] ("At step 106, the packet is accepted, if the plurality of objects satisfies the set of criteria. For example, the set of criteria may include a set criterion Such that the packets having the TCP flags syn and ack set, are accepted. Further, the set of criteria may include a not-set criterion, for example, the packets having TCP flag fin not set are accepted. In an embodiment of the invention, the accepted packet is transmitted to the network, if the accepted packet is an outgoing packet. In another embodiment of the inven tion, the accepted packet is transmitted to the network device, if the accepted packet is an incoming packet.").<br><br>Balakrishnan at [0022] ("FIG. 2 is a flowchart illustrating a method for matching objects in an application with the corresponding objects of a network device, in accordance with an exem-plary embodiment of the invention. In various embodiments of the invention, the matching of objects may be performed to manage network traffic in a network. The |

| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | application may be a packet of the network traffic. At step 202, a network device having a plurality of control list objects is provided. The control list objects may be, for example, filtering TCP packets in an Access Control List (ACL). The ACL is generally implemented as a data structure such as a tree that has the information (to be used by the network operating system) to determine the access rights of each packet. At step 204, at least one control list object of the network device is correspondingly matched with at least one object of the application. The matching is performed based on the objects being present or not present in the application. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG. 1. In an embodiment of the invention, a '+' sign is prefixed to a control list object in the 'set criterion', to signify that the packet is considered a match if the control list object is present in the packet. In another embodiment of the inven-tion, a'-' sign is prefixed to a control list object, to signify that the packet is considered a match if the control list object is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device..").<br><br>Balakrishnan atat [0023] ("FIG. 3 is a flowchart illustrating a method for determining if a packet is acceptable for transmitting to a network, in accordance with an exemplary embodiment of the invention. At step 302, a router including an ACL having match condition based on packet flags is provided. At step 304, a packet having packet flags is transmitted to an interface of the router that has an ACL. Subsequently, at least one packet flag of the packet is matched with at least one packet flag as specified in the ACL of the router, at step 306. The matching is performed based on the matching criteria for indicating that the packet is acceptable for transmitting to the network. If the packet flag( s) match the conditions in the ACL, then the packet is transmitted to the network. If the packet flag( s) do not satisfy the conditions in the ACL, the packet is not transmitted. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG. 1. In an embodiment of the invention, a'+' sign is prefixed to an access flag in the 'set criterion', to signify that the packet is considered a match if the access flag is present in the packet. In another embodiment of the invention, a '-' sign is prefixed to an access flag, to signify that the packet is considered a match if the access flag is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device.") |

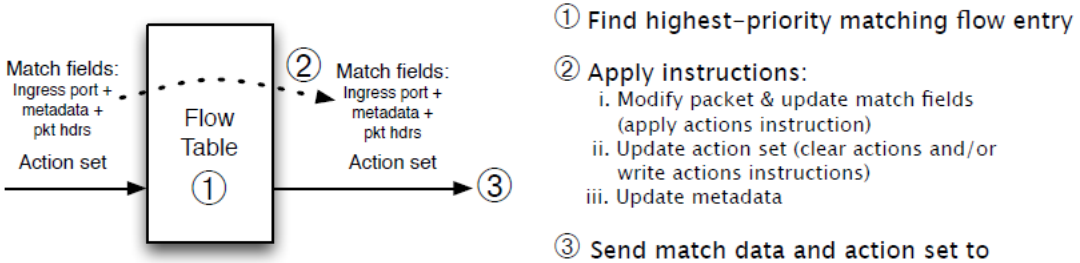
| No. | '111 Patent Claim 1 | The Reference |
|---|---|---|
| | | Balakrishnan atat [0040] ("FIG. 4 is a flowchart illustrating a method for determining if a packet is allowed to transit a router, in accordance with an exemplary embodiment of the invention. At step 402, a router including an ACL having access flags is provided. At step 404, a packet having packet flags is received via an interface of the router that has an ACL. Subsequently, at least one packet flag of the packet is matched in a corresponding one-to-one relationship with at least one access flag of the ACL of the router, at step 406. The matching is performed based on the matching criteria for indicating that the packet is acceptable for transmitting to the router. If the packet flag(s) and the access flag(s) match, then the packet is transmitted to the router. If the packet flag(s) and the access flag(s) do not match, the packet is not transmitted. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG.1. In an embodiment of the invention, a '+' sign is prefixed to an access flag in the 'set criterion', to signify that the packet is considered a match if the access flag is present in the packet. In another embodiment of the invention, a'-' sign is prefixed to an access flag, to signify that the packet is considered a match if the access flag is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device..").<br><br> |

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|
| 2[a] | The method according to claim 1, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction, and | The Reference discloses the method according to claim 1, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 11 |

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|




(a) Packets are matched against multiple tables in the pipeline




(b) Per-table packet processing

Figure 2: Packet flow through the processing pipeline

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|
| | | The flow tables of an OpenFlow switch are sequentially numbered, starting at 0. Pipeline processing always starts at the first flow table: the packet is first matched against flow entries of flow table 0. Other flow tables may be used depending on the outcome of the match in the first table.<br><br>When processed by a flow table, the packet is matched against the flow entries of the flow table to select a flow entry (see 5.3). If a flow entry is found, the instruction set included in that flow entry is executed, those instructions may explicitly direct the packet to another flow table (using the Goto Instruction, see 5.9), where the same process is repeated again. A flow entry can only direct a packet to a flow table number which is greater than its own flow table number, in other words pipeline processing can only go forward and not backward. Obviously, the flow entries of the last table of the pipeline can not include the Goto instruction. If the matching flow entry does not direct packets to another flow table, pipeline processing stops at this table. When pipeline processing stops, the packet is processed with its associated action set and usually forwarded (see 5.10).<br><br>If a packet does not match a flow entry in a flow table, this is a table miss. The behavior on a table miss depends on the table configuration (see 5.4). A table-miss flow entry in the flow table may specify how to process unmatched packets: Options include dropping them, passing them to another table or sending them to the controller over the control channel via packet-in messages (see 6.1.2).<br><br>OpenFlow at 12 |

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

Table 1: Main components of a flow entry in a flow table.

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **priority**: matching precedence of the flow entry
- **counters**: to update for matching packets
- **instructions** to modify the action set or pipeline processing
- **timeouts**: maximum amount of time or idle time before flow is expired by the switch
- **cookie**: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion, not used when processing packets.

A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table. The flow entry that wildcards all fields (all fields omitted) and has priority equal 0 is called the table-miss flow entry (see 5.4).

## 5.3 Matching

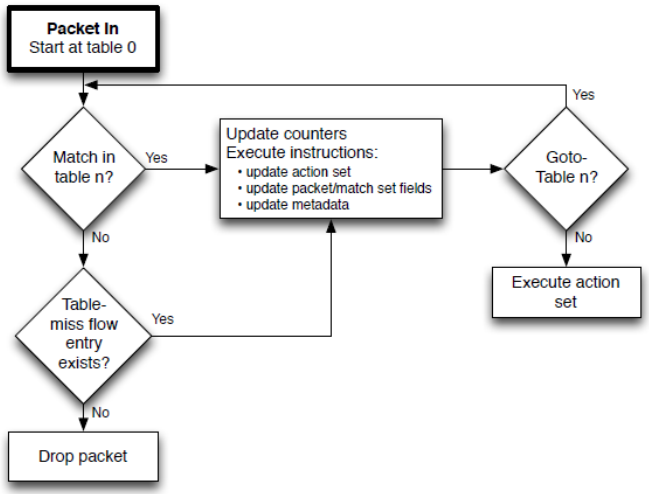



Figure 3: Flowchart detailing packet flow through an OpenFlow switch.

On receipt of a packet, an OpenFlow Switch performs the functions shown in Figure 3. The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables (see 5.1).

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|
| | | OpenFlow at 13<br><br>**5.4 Table-miss**<br><br>Every flow table must support a table-miss flow entry to process table misses. The table-miss flow entry specifies how to process packets unmatched by other flow entries in the flow table (see 5.1), and may, for example send packets to the controller, drop packets or direct packets to a subsequent table.<br><br>The table-miss flow entry is identified by its match and its priority (see 5.2), it wildcards all match fields (all fields omitted) and has the lowest priority (0). The match of the table-miss flow entry may fall outside the normal range of matches supported by a flow table, for example an exact match table would not support wildcards for other flow entries but must support the table-miss flow entry wildcarding all fields. The table-miss flow entry may not have the same capability as regular flow entry (see A.3.5.5). Implementations are encouraged to support for table-miss flow entries at minimum the same capability as the table-miss processing of previous versions of OpenFlow: send packets to the controller, drop packets or direct packets to a subsequent table.<br><br>The table-miss flow entry behave in most ways like any other flow entry : it does not exist by default in a flow table, the controller may add it or remove it at any time (see 6.4), and it may expire (see 5.5). The table-miss flow entry matches packets in the table as expected from its set of match fields and priority (see 5.3), it matches packets unmatched by other flow entries in the flow table. The table-miss flow entry instructions are applied to packets matching the table-miss flow entry (see 5.9). If the table-miss flow entry directly sends packets to the controller using the CONTROLLER port (see 4.5), the packet-in reason must identify a table-miss (see A.4.1).<br><br>If the table-miss flow entry does not exist, by default packets unmatched by flow entries are dropped (discarded). A switch configuration, for example using the OpenFlow Configuration Protocol, may override this default and specify another behaviour. |
| 2[b] | upon receiving by the network node the 'terminate ' instruction, the method further comprising blocking, by the network node, the packet from being sent to the second entity and to the controller. | The Reference discloses upon receiving by the network node the 'terminate ' instruction, the method further comprising blocking, by the network node, the packet from being sent to the second entity and to the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, and Balakrishnan. |

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|
| | | Below are examples of such references.<br><br>OpenFlow at 13<br><br>**5.4 Table-miss**<br><br>Every flow table must support a table-miss flow entry to process table misses. The table-miss flow entry specifies how to process packets unmatched by other flow entries in the flow table (see 5.1), and may, for example send packets to the controller, drop packets or direct packets to a subsequent table.<br><br>The table-miss flow entry is identified by its match and its priority (see 5.2), it wildcards all match fields (all fields omitted) and has the lowest priority (0). The match of the table-miss flow entry may fall outside the normal range of matches supported by a flow table, for example an exact match table would not support wildcards for other flow entries but must support the table-miss flow entry wildcarding all fields. The table-miss flow entry may not have the same capability as regular flow entry (see A.3.5.5). Implementations are encouraged to support for table-miss flow entries at minimum the same capability as the table-miss processing of previous versions of OpenFlow: send packets to the controller, drop packets or direct packets to a subsequent table.<br><br>The table-miss flow entry behave in most ways like any other flow entry : it does not exist by default in a flow table, the controller may add it or remove it at any time (see 6.4), and it may expire (see 5.5). The table-miss flow entry matches packets in the table as expected from its set of match fields and priority (see 5.3), it matches packets unmatched by other flow entries in the flow table. The table-miss flow entry instructions are applied to packets matching the table-miss flow entry (see 5.9). If the table-miss flow entry directly sends packets to the controller using the CONTROLLER port (see 4.5), the packet-in reason must identify a table-miss (see A.4.1).<br><br>If the table-miss flow entry does not exist, by default packets unmatched by flow entries are dropped (discarded). A switch configuration, for example using the OpenFlow Configuration Protocol, may override this default and specify another behaviour.<br><br>Balakrishnan at [0020] ("In an embodiment of the invention, a '+' sign is prefixed to a packet flag in the 'set criterion', to signify that the packet is considered a match if the packet flag is present in the packet. In another embodiment of the invention, a '−' sign is prefixed to a packet flag, to signify that the packet is considered a match if the packet flag is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device.").<br><br>Balakrishnan at [0021] ("At step 106, the packet is accepted, if the plurality of objects satisfies the set of criteria. For example, the set of criteria may include a set criterion Such that the packets having the TCP flags syn and ack set, are accepted. Further, the set of |

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|
| | | criteria may include a not-set criterion, for example, the packets having TCP flag fin not set are accepted. In an embodiment of the invention, the accepted packet is transmitted to the network, if the accepted packet is an outgoing packet. In another embodiment of the inven tion, the accepted packet is transmitted to the network device, if the accepted packet is an incoming packet.").<br><br>Balakrishnan at [0022] ("FIG. 2 is a flowchart illustrating a method for matching objects in an application with the corresponding objects of a network device, in accordance with an exem-plary embodiment of the invention. In various embodiments of the invention, the matching of objects may be performed to manage network traffic in a network. The application may be a packet of the network traffic. At step 202, a network device having a plurality of control list objects is provided. The control list objects may be, for example, filtering TCP packets in an Access Control List (ACL). The ACL is generally implemented as a data structure such as a tree that has the information (to be used by the network operating system) to determine the access rights of each packet. At step 204, at least one control list object of the network device is correspondingly matched with at least one object of the application. The matching is performed based on the objects being present or not present in the application. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG. 1. In an embodiment of the invention, a '+' sign is prefixed to a control list object in the 'set criterion', to signify that the packet is considered a match if the control list object is present in the packet. In another embodiment of the inven-tion, a'-' sign is prefixed to a control list object, to signify that the packet is considered a match if the control list object is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device..").<br><br>Balakrishnan atat [0023] ("FIG. 3 is a flowchart illustrating a method for determining if a packet is acceptable for transmitting to a network, in accordance with an exemplary embodiment of the invention. At step 302, a router including an ACL having match condition based on packet flags is provided. At step 304, a packet having packet flags is transmitted to an interface of the router that has an ACL. Subsequently, at least one packet flag of the packet is matched with at least one packet flag as specified in the ACL of the router, at step 306. The matching is performed based on the matching criteria for indicating that the packet is acceptable for transmitting to the network. If the packet flag( s) match the |

| No. | '111 Patent Claim 2 | The Reference |
|---|---|---|
| | | conditions in the ACL, then the packet is transmitted to the network. If the packet flag( s) do not satisfy the conditions in the ACL, the packet is not transmitted. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG. 1. In an embodiment of the invention, a'+' sign is prefixed to an access flag in the 'set criterion', to signify that the packet is considered a match if the access flag is present in the packet. In another embodiment of the invention, a '-' sign is prefixed to an access flag, to signify that the packet is considered a match if the access flag is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device.").<br><br>Balakrishnan atat [0040] ("FIG. 4 is a flowchart illustrating a method for determining if a packet is allowed to transit a router, in accordance with an exemplary embodiment of the invention. At step 402, a router including an ACL having access flags is provided. At step 404, a packet having packet flags is received via an interface of the router that has an ACL. Subsequently, at least one packet flag of the packet is matched in a corresponding one-to-one relationship with at least one access flag of the ACL of the router, at step 406. The matching is performed based on the matching criteria for indicating that the packet is acceptable for transmitting to the router. If the packet flag(s) and the access flag(s) match, then the packet is transmitted to the router. If the packet flag(s) and the access flag(s) do not match, the packet is not transmitted. The matching criteria may be at least one of a 'set criterion' or a 'not-set criterion', as described in conjunction with FIG.1. In an embodiment of the invention, a '+' sign is prefixed to an access flag in the 'set criterion', to signify that the packet is considered a match if the access flag is present in the packet. In another embodiment of the invention, a'-' sign is prefixed to an access flag, to signify that the packet is considered a match if the access flag is not present in the packet. Once a packet matches the criteria specified, the packet can be accepted or rejected based on the policy of the network device..") |

| No. | '111 Patent Claim 3 | The Reference |
|---|---|---|
| 3[a] | The method according to claim 1, wherein the instruction is a 'probe', a 'mirror', or | The Reference discloses the method according to claim 1, wherein the instruction is a 'probe', a 'mirror', or a 'terminate' instruction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was |

| No. | '111 Patent Claim 3 | The Reference |
|---|---|---|
| | a 'terminate' instruction, and | known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |
| 3[b] | upon receiving by the network node the 'mirror' instruction and responsive to the packet satisfying the criterion, method further comprising sending the packet, by the network node, to the second entity and to the controller. | The Reference discloses upon receiving by the network node the 'mirror' instruction and responsive to the packet satisfying the criterion, method further comprising sending the packet, by the network node, to the second entity and to the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| 4[a] | The method according to claim 1, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction, and | The Reference discloses the method according to claim 1, wherein the instruction is 'probe', 'mirror', or 'terminate' instruction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| 4[b] | upon receiving by the network node the 'probe' instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet, by the network node, to the controller; | The Reference discloses upon receiving by the network node the 'probe' instruction and responsive to the packet satisfying the criterion, the method further comprising: sending the packet, by the network node, to the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 22-23<br><br>**6.1.2 Asynchronous**<br><br>Asynchronous messages are sent without a controller soliciting them from a switch. Switches send asynchronous messages to controllers to denote a packet arrival, switch state change, or error. The four main asynchronous message types are described below.<br><br>**Packet-in:** Transfer the control of a packet to the controller. For all packets forwarded to the **CONTROLLER** reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers (see 5.12). Other processing, such as TTL checking, may also send packets to the controller using packet-in events. |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| | | Packet-in events can be configured to buffer packets. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2). If the packet-in event is configured to buffer packets and the switch has sufficient memory to buffer them, the packet-in events contain only some fraction of the packet header and a buffer ID to be used by a controller when it is ready for the switch to forward the packet. Switches that do not support internal buffering, are configured to not buffer packets for the packet-in event, or have run out of internal buffering, must send the full packet to controllers as part of the event. Buffered packets will usually be processed via a **Packet-out** message from a controller, or automatically expired after some time.<br><br>If the packet is buffered, the number of bytes of the original packet to include in the packet-in can be configured. By default, it is 128 bytes. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2).<br><br>**Flow-Removed:** Inform the controller about the removal of a flow entry from a flow table. Flow-Removed messages are only sent for flow entries with the `OFPFF_SEND_FLOW_REM` flag set. They are generated as the result of a controller flow delete requests or the switch flow expiry process when one of the flow timeout is exceeded (see 5.5).<br><br>**Port-status:** Inform the controller of a change on a port. The switch is expected to send port-status messages to controllers as port configuration or port state changes. These events include change in port configuration events, for example if it was brought down directly by a user, and port state change events, for example if the link went down.<br><br>**Error:** The switch is able to notify controllers of problems using error messages.<br><br>OpenFlow at 36 |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| | | The port numbers use the following conventions:<br><br>`/* Port numbering. Ports are numbered starting from 1. */`<br>`enum ofp_port_no {`<br>`  /* Maximum number of physical and logical switch ports. */`<br>`  OFPP_MAX  = 0xffffff00,`<br><br>`  /* Reserved OpenFlow Port (fake output "ports"). */`<br>`  OFPP_IN_PORT  = 0xfffffff8,  /* Send the packet out the input port. This reserved port must be explicitly used in order to send back out of the input port. */`<br>`  OFPP_TABLE  = 0xfffffff9,  /* Submit the packet to the first flow table NB: This destination port can only be used in packet-out messages. */`<br>`  OFPP_NORMAL  = 0xfffffffa,  /* Process with normal L2/L3 switching. */`<br>`  OFPP_FLOOD  = 0xfffffffb,  /* All physical ports in VLAN, except input port and those blocked or link down. */`<br>`  OFPP_ALL  = 0xfffffffc,  /* All physical ports except input port. */`<br>`  OFPP_CONTROLLER = 0xfffffffd,  /* Send to controller. */`<br>`  OFPP_LOCAL  = 0xfffffffe,  /* Local openflow "port". */`<br><br>OpenFlow at 76-77 |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|

## A.4 Asynchronous Messages

### A.4.1 Packet-In Message

When packets are received by the datapath and sent to the controller, they use the `OFPT_PACKET_IN` message:

```
/* Packet received on port (datapath -> controller). */
struct ofp_packet_in {
    struct ofp_header header;
    uint32_t buffer_id;     /* ID assigned by datapath. */
    uint16_t total_len;     /* Full length of frame. */
    uint8_t reason;         /* Reason packet is being sent (one of OFPR_*) */
    uint8_t table_id;       /* ID of the table that was looked up */
    uint64_t cookie;        /* Cookie of the flow entry that was looked up. */
    struct ofp_match match; /* Packet metadata. Variable size. */
    /* Followed by:
     *   - Exactly 2 all-zero padding bytes, then
     *   - An Ethernet frame whose length is inferred from header.length.
     * The padding bytes preceding the Ethernet frame ensure that the IP
     * header (if any) following the Ethernet header is 32-bit aligned.
     */
    //uint8_t pad[2];       /* Align to 64 bit + 16 bit */
    //uint8_t data[0];      /* Ethernet frame */
};
OFP_ASSERT(sizeof(struct ofp_packet_in) == 32);
```

The `buffer_id` is an opaque value used by the datapath to identify a buffered packet. When a packet is buffered, some number of bytes from the message will be included in the data portion of the message. If the packet is sent because of a "send to controller" action, then `max_len` bytes from the `ofp_action_output` of the flow setup request are sent. If the packet is sent for other reasons, such as an invalid TTL, then at least `miss_send_len` bytes from the `OFPT_SET_CONFIG` message are sent. The default `miss_send_len` is 128 bytes. If the packet is not buffered - either because of no available buffers, or because of explicitly requested via `OFPCML_NO_BUFFER` - the entire packet is included in the data portion, and the `buffer_id` is `OFP_NO_BUFFER`.

Switches that implement buffering are expected to expose, through documentation, both the amount of available buffering, and the length of time before buffers may be reused. A switch must gracefully handle

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|

the case where a buffered `packet_in` message yields no response from the controller. A switch should prevent a buffer from being reused until it has been handled by the controller, or some amount of time (indicated in documentation) has passed.

The `data` field contains the packet itself, or a fraction of the packet if the packet is buffered. The packet header reflect any changes applied to the packet in previous processing.

The `reason` field can be any of these values:

```
/* Why is this packet being sent to the controller? */
enum ofp_packet_in_reason {
    OFPR_NO_MATCH    = 0,   /* No matching flow (table-miss flow entry). */
    OFPR_ACTION      = 1,   /* Action explicitly output to controller. */
    OFPR_INVALID_TTL = 2,   /* Packet has invalid TTL */
};
```

`OFPR_INVALID_TTL` indicates that a packets with an invalid IP TTL or MPLS TTL was rejected by the OpenFlow pipeline and passed to the controller. Checking for invalid TTL does not need to be done for every packet, but it must be done at a minimum every time a `OFPAT_DEC_MPLS_TTL` or `OFPAT_DEC_NW_TTL` action is applied to a packet.

The `cookie` field contains the cookie of the flow entry that caused the packet to be sent to the controller. This field must be set to -1 (0xffffffffffffffff) if a cookie cannot be associated with a particular flow. For example, if the packet-in was generated in a group bucket or from the action set.

The `match` field reflect the packet's headers and context when the event that triggers the packet-in message occurred and contains a set of OXM TLVs. This context includes any changes applied to the packet in previous processing, including actions already executed, if any, but not any changes in the action set. The OXM TLVs must include context fields, that is, fields whose values cannot be determined from the packet data. The standard context fields are `OFPXMT_OFB_IN_PORT`, `OFPXMT_OFB_IN_PHY_PORT`, `OFPXMT_OFB_METADATA` and `OFPXMT_OFB_TUNNEL_ID`. Fields whose values are all-bits-zero may be omitted. Optionally, the OXM TLVs may also include packet header fields that were previously extracted from the packet, including any modifications of those in the course of the processing.

When a packet is received directly on a physical port and not processed by a logical port, `OFPXMT_OFB_IN_PORT` and `OFPXMT_OFB_IN_PHY_PORT` have the same value, the OpenFlow port_no of this physical port. `OFPXMT_OFB_IN_PHY_PORT` may be omitted if it has the same value as `OFPXMT_OFB_IN_PORT`.

When a packet is received on a logical port by way of a physical port, `OFPXMT_OFB_IN_PORT` is the logical port's port_no and `OFPXMT_OFB_IN_PHY_PORT` is the physical port's port_no. For example, consider a packet received on a tunnel interface defined over a link aggregation group (LAG) with two physical port members. If the tunnel interface is the logical port bound to OpenFlow, then `OFPXMT_OFB_IN_PORT` is the tunnel port_no and `OFPXMT_OFB_IN_PHY_PORT` is the physical port_no member of the LAG on which the tunnel is configured.

The port referenced by the `OFPXMT_OFB_IN_PORT` TLV must be the port used for matching flow entries (see 5.3) and must be available to OpenFlow processing (i.e. OpenFlow can forward packet to this port, depending on port flags). `OFPXMT_OFB_IN_PHY_PORT` need not be available for matching or OpenFlow processing.

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| 4[c] | responsive to receiving the packet, analyzing the packet, by the controller; | The Reference discloses responsive to receiving the packet, analyzing the packet, by the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |
| 4[d] | sending the packet, by the controller, to the network node; and | The Reference discloses sending the packet, by the controller, to the network node.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 22<br><br>**Packet-out**: These are used by the controller to send packets out of a specified port on the switch, and to forward packets received via Packet-in messages. Packet-out messages must contain a full packet or a buffer ID referencing a packet stored in the switch. The message must also contain a list of actions to be applied in the order they are specified; an empty action list drops the packet.<br><br>OpenFlow at 74 |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| | | **A.3.7 Packet-Out Message**<br><br>When the controller wishes to send a packet out through the datapath, it uses the `OFPT_PACKET_OUT` message:<br><br>`/* Send packet (controller -> datapath). */`<br>`struct ofp_packet_out {`<br>`  struct ofp_header header;`<br>`  uint32_t buffer_id;  /* ID assigned by datapath (OFP_NO_BUFFER if none). */`<br>`  uint32_t in_port;  /* Packet's input port or OFPP_CONTROLLER. */`<br>`  uint16_t actions_len;  /* Size of action array in bytes. */`<br>`  uint8_t pad[6];`<br>`  struct ofp_action_header actions[0]; /* Action list. */`<br>`  /* uint8_t data[0]; */  /* Packet data. The length is inferred from the length field in the header. (Only meaningful if buffer_id == -1.) */`<br>`};`<br>`OFP_ASSERT(sizeof(struct ofp_packet_out) == 24);`<br><br>The `buffer_id` is the same given in the `ofp_packet_in` message. If the `buffer_id` is `OFP_NO_BUFFER`, then the packet data is included in the data array.<br><br>The `in_port` field is the ingress port that must be associated with the packet for OpenFlow processing. It must be set to either a valid standard switch port or `OFPP_CONTROLLER`.<br><br>The `action` field is an action list defining how the packet should be processed by the switch. It may include packet modification, group processing and an output port. The action list of an `OFPT_PACKET_OUT` message can also specify the `OFPP_TABLE` reserved port as an output action to process the packet through the existing flow entries, starting at the first flow table. If `OFPP_TABLE` is specified, the `in_port` field is used as the ingress port in the flow table lookup.<br><br>Packets sent to `OFPP_TABLE` may be forwarded back to the controller as the result of a flow entry action or table miss. Detecting and taking action for such controller-to-switch loops is outside the scope of this specification. In general, OpenFlow messages are not guaranteed to be processed in order, therefore if a `OFPT_PACKET_OUT` message using `OFPP_TABLE` depends on a flow that was recently sent to the switch (with a `OFPT_FLOW_MOD` message), a `OFPT_BARRIER_REQUEST` message may be required prior to the `OFPT_PACKET_OUT` message to make sure the flow entry was committed to the flow table prior to execution of `OFPP_TABLE`. |
| 4[e] | responsive to receiving the packet, sending the packet, by the network node, to the second entity. | The Reference discloses responsive to receiving the packet, sending the packet, by the network node, to the second entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| | | skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 22<br><br>**Packet-out**: These are used by the controller to send packets out of a specified port on the switch, and to forward packets received via Packet-in messages. Packet-out messages must contain a full packet or a buffer ID referencing a packet stored in the switch. The message must also contain a list of actions to be applied in the order they are specified; an empty action list drops the packet.<br><br>OpenFlow at 74 |

| No. | '111 Patent Claim 4 | The Reference |
|---|---|---|
| | | **A.3.7 Packet-Out Message**<br><br>When the controller wishes to send a packet out through the datapath, it uses the `OFPT_PACKET_OUT` message:<br><br>`/* Send packet (controller -> datapath). */`<br>`struct ofp_packet_out {`<br>`  struct ofp_header header;`<br>`  uint32_t buffer_id;  /* ID assigned by datapath (OFP_NO_BUFFER if none). */`<br>`  uint32_t in_port;  /* Packet's input port or OFPP_CONTROLLER. */`<br>`  uint16_t actions_len;  /* Size of action array in bytes. */`<br>`  uint8_t pad[6];`<br>`  struct ofp_action_header actions[0]; /* Action list. */`<br>`  /* uint8_t data[0]; */  /* Packet data. The length is inferred from the length field in the header. (Only meaningful if buffer_id == -1.) */`<br>`};`<br>`OFP_ASSERT(sizeof(struct ofp_packet_out) == 24);`<br><br>The `buffer_id` is the same given in the `ofp_packet_in` message. If the `buffer_id` is `OFP_NO_BUFFER`, then the packet data is included in the data array.<br><br>The `in_port` field is the ingress port that must be associated with the packet for OpenFlow processing. It must be set to either a valid standard switch port or `OFPP_CONTROLLER`.<br><br>The `action` field is an action list defining how the packet should be processed by the switch. It may include packet modification, group processing and an output port. The action list of an `OFPT_PACKET_OUT` message can also specify the `OFPP_TABLE` reserved port as an output action to process the packet through the existing flow entries, starting at the first flow table. If `OFPP_TABLE` is specified, the `in_port` field is used as the ingress port in the flow table lookup.<br><br>Packets sent to `OFPP_TABLE` may be forwarded back to the controller as the result of a flow entry action or table miss. Detecting and taking action for such controller-to-switch loops is outside the scope of this specification. In general, OpenFlow messages are not guaranteed to be processed in order, therefore if a `OFPT_PACKET_OUT` message using `OFPP_TABLE` depends on a flow that was recently sent to the switch (with a `OFPT_FLOW_MOD` message), a `OFPT_BARRIER_REQUEST` message may be required prior to the `OFPT_PACKET_OUT` message to make sure the flow entry was committed to the flow table prior to execution of `OFPP_TABLE`. |

| No. | '111 Patent Claim 5 | The Reference |
|---|---|---|
| 5 | The method according to claim 1, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof, by the network node, to the controller. | The Reference discloses the method according to claim 1, further comprising responsive to the packet satisfying the criterion and to the instruction, sending the packet or a portion thereof, by the network node, to the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 22-23<br><br>**6.1.2 Asynchronous**<br><br>Asynchronous messages are sent without a controller soliciting them from a switch. Switches send asynchronous messages to controllers to denote a packet arrival, switch state change, or error. The four main asynchronous message types are described below.<br><br>**Packet-in:** Transfer the control of a packet to the controller. For all packets forwarded to the **CONTROLLER** reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers (see 5.12). Other processing, such as TTL checking, may also send packets to the controller using packet-in events. |

| No. | '111 Patent Claim 5 | The Reference |
|---|---|---|
| | | Packet-in events can be configured to buffer packets. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2). If the packet-in event is configured to buffer packets and the switch has sufficient memory to buffer them, the packet-in events contain only some fraction of the packet header and a buffer ID to be used by a controller when it is ready for the switch to forward the packet. Switches that do not support internal buffering, are configured to not buffer packets for the packet-in event, or have run out of internal buffering, must send the full packet to controllers as part of the event. Buffered packets will usually be processed via a **Packet-out** message from a controller, or automatically expired after some time.<br><br>If the packet is buffered, the number of bytes of the original packet to include in the packet-in can be configured. By default, it is 128 bytes. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2).<br><br>**Flow-Removed:** Inform the controller about the removal of a flow entry from a flow table. Flow-Removed messages are only sent for flow entries with the `OFPFF_SEND_FLOW_REM` flag set. They are generated as the result of a controller flow delete requests or the switch flow expiry process when one of the flow timeout is exceeded (see 5.5).<br><br>**Port-status:** Inform the controller of a change on a port. The switch is expected to send port-status messages to controllers as port configuration or port state changes. These events include change in port configuration events, for example if it was brought down directly by a user, and port state change events, for example if the link went down.<br><br>**Error:** The switch is able to notify controllers of problems using error messages.<br><br>OpenFlow at 36 |

| No. | '111 Patent Claim 5 | The Reference |
|---|---|---|
| | | The port numbers use the following conventions:<br><br>`/* Port numbering. Ports are numbered starting from 1. */`<br>`enum ofp_port_no {`<br>`  /* Maximum number of physical and logical switch ports. */`<br>`  OFPP_MAX  = 0xffffff00,`<br><br>`  /* Reserved OpenFlow Port (fake output "ports"). */`<br>`  OFPP_IN_PORT  = 0xfffffff8,  /* Send the packet out the input port. This reserved port must be explicitly used in order to send back out of the input port. */`<br>`  OFPP_TABLE  = 0xfffffff9,  /* Submit the packet to the first flow table NB: This destination port can only be used in packet-out messages. */`<br>`  OFPP_NORMAL  = 0xfffffffa,  /* Process with normal L2/L3 switching. */`<br>`  OFPP_FLOOD  = 0xfffffffb,  /* All physical ports in VLAN, except input port and those blocked or link down. */`<br>`  OFPP_ALL  = 0xfffffffc,  /* All physical ports except input port. */`<br>`  OFPP_CONTROLLER = 0xfffffffd,  /* Send to controller. */`<br>`  OFPP_LOCAL  = 0xfffffffe,  /* Local openflow "port". */` |

| No. | '111 Patent Claim 6 | The Reference |
|---|---|---|
| 6 | The method according to claim 5, further comprising storing the received packet or a portion thereof, by the controller, in a memory. | The Reference discloses the method according to claim 5, further comprising storing the received packet or a portion thereof, by the controller, in a memory.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 7 | The Reference |
|---|---|---|
| 7 | The method according to claim 5, further comprising responsive to the packet satisfying the criterion and to instruction, sending a portion of the packet, by the network node, to the controller. | The Reference discloses the method according to claim 5, further comprising responsive to the packet satisfying the criterion and to instruction, sending a portion of the packet, by the network node, to the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 22-23<br><br>**6.1.2 Asynchronous**<br><br>Asynchronous messages are sent without a controller soliciting them from a switch. Switches send asynchronous messages to controllers to denote a packet arrival, switch state change, or error. The four main asynchronous message types are described below.<br><br>**Packet-in:** Transfer the control of a packet to the controller. For all packets forwarded to the **CONTROLLER** reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers (see 5.12). Other processing, such as TTL checking, may also send packets to the controller using packet-in events. |

| No. | '111 Patent Claim 7 | The Reference |
|---|---|---|
| | | Packet-in events can be configured to buffer packets. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2). If the packet-in event is configured to buffer packets and the switch has sufficient memory to buffer them, the packet-in events contain only some fraction of the packet header and a buffer ID to be used by a controller when it is ready for the switch to forward the packet. Switches that do not support internal buffering, are configured to not buffer packets for the packet-in event, or have run out of internal buffering, must send the full packet to controllers as part of the event. Buffered packets will usually be processed via a **Packet-out** message from a controller, or automatically expired after some time.<br><br>If the packet is buffered, the number of bytes of the original packet to include in the packet-in can be configured. By default, it is 128 bytes. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2).<br><br>**Flow-Removed:** Inform the controller about the removal of a flow entry from a flow table. Flow-Removed messages are only sent for flow entries with the `OFPFF_SEND_FLOW_REM` flag set. They are generated as the result of a controller flow delete requests or the switch flow expiry process when one of the flow timeout is exceeded (see 5.5).<br><br>**Port-status:** Inform the controller of a change on a port. The switch is expected to send port-status messages to controllers as port configuration or port state changes. These events include change in port configuration events, for example if it was brought down directly by a user, and port state change events, for example if the link went down.<br><br>**Error:** The switch is able to notify controllers of problems using error messages. |

| No. | '111 Patent Claim 8 | The Reference |
|---|---|---|
| 8[a] | The method according to claim 7, wherein the portion of<br>the packet consists of multiple consecutive bytes, and | The Reference discloses the method according to claim 7, wherein the portion of the packet consists of multiple consecutive bytes.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br>Below are examples of such references. |

| No. | '111 Patent Claim 8 | The Reference |
|---|---|---|
| | | OpenFlow at 22-23<br>**6.1.2 Asynchronous**<br>Asynchronous messages are sent without a controller soliciting them from a switch. Switches send asynchronous messages to controllers to denote a packet arrival, switch state change, or error. The four main asynchronous message types are described below.<br>**Packet-in:** Transfer the control of a packet to the controller. For all packets forwarded to the **CONTROLLER** reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers (see 5.12). Other processing, such as TTL checking, may also send packets to the controller using packet-in events.<br>Packet-in events can be configured to buffer packets. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2). If the packet-in event is configured to buffer packets and the switch has sufficient memory to buffer them, the packet-in events contain only some fraction of the packet header and a buffer ID to be used by a controller when it is ready for the switch to forward the packet. Switches that do not support internal buffering, are configured to not buffer packets for the packet-in event, or have run out of internal buffering, must send the full packet to controllers as part of the event. Buffered packets will usually be processed via a **Packet-out** message from a controller, or automatically expired after some time.<br>If the packet is buffered, the number of bytes of the original packet to include in the packet-in can be configured. By default, it is 128 bytes. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2).<br>**Flow-Removed:** Inform the controller about the removal of a flow entry from a flow table. Flow-Removed messages are only sent for flow entries with the `OFPFF_SEND_FLOW_REM` flag set. They are generated as the result of a controller flow delete requests or the switch flow expiry process when one of the flow timeout is exceeded (see 5.5).<br>**Port-status:** Inform the controller of a change on a port. The switch is expected to send port-status messages to controllers as port configuration or port state changes. These events include change in port configuration events, for example if it was brought down directly by a user, and port state change events, for example if the link went down.<br>**Error:** The switch is able to notify controllers of problems using error messages. |
| 8[b] | wherein the instruction comprises | The Reference discloses wherein the instruction comprises identification of the consecutive bytes in the packet. |

| No. | '111 Patent Claim 8 | The Reference |
|---|---|---|
| | identification of the consecutive bytes in the packet. | To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 22-23<br><br>**6.1.2 Asynchronous**<br><br>Asynchronous messages are sent without a controller soliciting them from a switch. Switches send asynchronous messages to controllers to denote a packet arrival, switch state change, or error. The four main asynchronous message types are described below.<br><br>**Packet-in:** Transfer the control of a packet to the controller. For all packets forwarded to the **CONTROLLER** reserved port using a flow entry or the table-miss flow entry, a packet-in event is always sent to controllers (see 5.12). Other processing, such as TTL checking, may also send packets to the controller using packet-in events. |

| No. | '111 Patent Claim 8 | The Reference |
|---|---|---|
| | | Packet-in events can be configured to buffer packets. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2). If the packet-in event is configured to buffer packets and the switch has sufficient memory to buffer them, the packet-in events contain only some fraction of the packet header and a buffer ID to be used by a controller when it is ready for the switch to forward the packet. Switches that do not support internal buffering, are configured to not buffer packets for the packet-in event, or have run out of internal buffering, must send the full packet to controllers as part of the event. Buffered packets will usually be processed via a **Packet-out** message from a controller, or automatically expired after some time.<br><br>If the packet is buffered, the number of bytes of the original packet to include in the packet-in can be configured. By default, it is 128 bytes. For packet-in generated by an output action in a flow entries or group bucket, it can be specified individually in the output action itself (see A.2.5), for other packet-in it can be configured in the switch configuration (see A.3.2).<br><br>**Flow-Removed:** Inform the controller about the removal of a flow entry from a flow table. Flow-Removed messages are only sent for flow entries with the `OFPFF_SEND_FLOW_REM` flag set. They are generated as the result of a controller flow delete requests or the switch flow expiry process when one of the flow timeout is exceeded (see 5.5).<br><br>**Port-status:** Inform the controller of a change on a port. The switch is expected to send port-status messages to controllers as port configuration or port state changes. These events include change in port configuration events, for example if it was brought down directly by a user, and port state change events, for example if the link went down.<br><br>**Error:** The switch is able to notify controllers of problems using error messages. |

| No. | '111 Patent Claim 9 | The Reference |
|---|---|---|
| 9 | The method according to claim 5, further comprising responsive to receiving the packet, analyzing the packet, by the controller. | The Reference discloses the method according to claim 5, further comprising responsive to receiving the packet, analyzing the packet, by the controller.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 12 | The Reference |
|---|---|---|
| 12 | The method according to claim 9, wherein the analyzing comprises applying security or data analytic application. | The Reference discloses the method according to claim 9, wherein the analyzing comprises applying security or data analytic application.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 13 | The Reference |
|---|---|---|
| 13 | The method according to claim 9, wherein the analyzing comprises applying security application that comprises firewall or intrusion detection functionality. | The Reference discloses the method according to claim 9, wherein the analyzing comprises applying security application that comprises firewall or intrusion detection functionality.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 14 | The Reference |
|---|---|---|
| 14 | The method according to claim 9, wherein the analyzing comprises performing Deep Packet Inspection (DPI) or using a DPI engine on the packet. | The Reference discloses the method according to claim 9, wherein the analyzing comprises performing Deep Packet Inspection (DPI) or using a DPI engine on the packet.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 15 | The Reference |
|---|---|---|
| 15[a] | The method according to claim 9, wherein the packet comprises distinct header and payload fields, and | The Reference discloses the method according to claim 9, wherein the packet comprises distinct header and payload fields.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 12 |

| No. | '111 Patent Claim 15 | The Reference |
|---|---|---|

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

Table 1: Main components of a flow entry in a flow table.

- **match fields**: to match against packets. These consist of the ingress port and packet headers, and optionally metadata specified by a previous table.
- **priority**: matching precedence of the flow entry
- **counters**: to update for matching packets
- **instructions** to modify the action set or pipeline processing
- **timeouts**: maximum amount of time or idle time before flow is expired by the switch
- **cookie**: opaque data value chosen by the controller. May be used by the controller to filter flow statistics, flow modification and flow deletion, not used when processing packets.

A flow table entry is identified by its match fields and priority: the match fields and priority taken together identify a unique flow entry in the flow table. The flow entry that wildcards all fields (all fields omitted) and has priority equal 0 is called the table-miss flow entry (see 5.4).

## 5.3 Matching

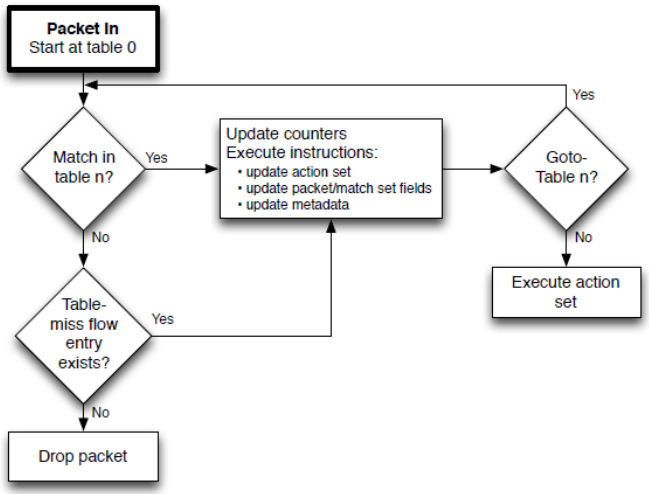



Figure 3: Flowchart detailing packet flow through an OpenFlow switch.

On receipt of a packet, an OpenFlow Switch performs the functions shown in Figure 3. The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables (see 5.1).

| No. | '111 Patent Claim 15 | The Reference |
|---|---|---|
| | | OpenFlow at 13<br><br>Packet match fields are extracted from the packet. Packet match fields used for table lookups depend on the packet type, and typically include various packet header fields, such as Ethernet source address or IPv4 destination address (see A.2.3). In addition to packet headers, matches can also be performed against the ingress port and metadata fields. Metadata may be used to pass information between tables in a switch. The packet match fields represent the packet in its current state, if actions applied in a previous table using the *Apply-Actions* changed the packet headers, those changes are reflected in the packet match fields.<br><br>A packet matches a flow table entry if the values in the packet match fields used for the lookup match those defined in the flow table entry. If a flow table entry field has a value of ANY (field omitted), it matches all possible values in the header. If the switch supports arbitrary bitmasks on specific match fields, these masks can more precisely specify matches.<br><br>The packet is matched against the table and *only* the highest priority flow entry that matches the packet must be selected. The counters associated with the selected flow entry must be updated and the instruction set included in the selected flow entry must be applied. If there are multiple matching flow entries with the same highest priority, the selected flow entry is explicitly undefined. This case can only arise when a controller writer never sets the `OFPFF_CHECK_OVERLAP` bit on flow mod messages and adds overlapping entries.<br><br>IP fragments must be reassembled before pipeline processing if the switch configuration contains the `OFPC_FRAG_REASM` flag (see A.3.2).<br><br>This version of the specification does *not* define the expected behavior when a switch receives a malformed or corrupted packet.<br><br>OpenFlow at 38-41<br><br>**A.2.3 Flow Match Structures**<br><br>An OpenFlow match is composed of a flow match header and a sequence of zero or more flow match fields. |

| No. | '111 Patent Claim 15 | The Reference |
|---|---|---|

### A.2.3.1 Flow Match Header

The flow match header is described by the `ofp_match` structure:

```
/* Fields to match against flows */
struct ofp_match {
    uint16_t type;              /* One of OFPMT_* */
    uint16_t length;            /* Length of ofp_match (excluding padding) */
    /* Followed by:
     *   - Exactly (length - 4) (possibly 0) bytes containing OXM TLVs, then
     *   - Exactly ((length + 7)/8*8 - length) (between 0 and 7) bytes of
     *     all-zero bytes
     * In summary, ofp_match is padded as needed, to make its overall size
     * a multiple of 8, to preserve alignement in structures using it.
     */
    uint8_t oxm_fields[4];      /* OXMs start here - Make compiler happy */
};
OFP_ASSERT(sizeof(struct ofp_match) == 8);
```

The `type` field is set to `OFPMT_OXM` and `length` field is set to the actual length of `ofp_match` structure including all match fields. The payload of the OpenFlow match is a set of OXM Flow match fields.

```
/* The match type indicates the match structure (set of fields that compose the
 * match) in use. The match type is placed in the type field at the beginning
 * of all match structures. The "OpenFlow Extensible Match" type corresponds
 * to OXM TLV format described below and must be supported by all OpenFlow
 * switches. Extensions that define other match types may be published on the
 * ONF wiki. Support for extensions is optional.
 */
enum ofp_match_type {
    OFPMT_STANDARD = 0,       /* Deprecated. */
    OFPMT_OXM      = 1,       /* OpenFlow Extensible Match */
};
```

The only valid match type in this specification is `OFPMT_OXM`, the OpenFlow 1.1 match type `OFPMT_STANDARD` is deprecated. If an alternate match type is used, the match fields and payload may be set differently, but this is outside the scope of this specification.

### A.2.3.2 Flow Match Field Structures

The flow match fields are described using the OpenFlow Extensible Match (OXM) format, which is a compact type-length-value (TLV) format. Each OXM TLV is 5 to 259 (inclusive) bytes long. OXM TLVs are not aligned on or padded to any multibyte boundary. The first 4 bytes of an OXM TLV are its header, followed by the entry's body.

An OXM TLV's header is interpreted as a 32-bit word in network byte order (see figure 4).

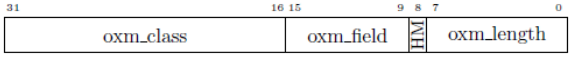



Figure 4: OXM TLV header layout

The OXM TLV's header fields are defined in Table 9

The `oxm_class` is a OXM match class that contains related match types, and is described in section A.2.3.3. `oxm_field` is an class-specific value, identifying one of the match types within the match class. The combination of `oxm_class` and `oxm_field` (the most-significant 23 bits of the header) are collectively `oxm_type`. The `oxm_type` normally designates a protocol header field, such as the Ethernet type, but it can also refer to packet metadata, such as the switch port on which a packet arrived.

| No. | '111 Patent Claim 15 | The Reference |
|---|---|---|

| Name | | Width | Usage |
|---|---|---|---|
| oxm_type | oxm_class | 16 | Match class: member class or reserved class |
| | oxm_field | 7 | Match field within the class |
| | oxm_hasmask | 1 | Set if OXM include a bitmask in payload |
| | oxm_length | 8 | Length of OXM payload |

Table 9: OXM TLV header fields

`oxm_hasmask` defines if the OXM TLV contains a bitmask, more details is explained in section A.2.3.5.

`oxm_length` is a positive integer describing the length of the OXM TLV payload in bytes. The length of the OXM TLV, including the header, is exactly 4 + `oxm_length` bytes.

For a given `oxm_class`, `oxm_field`, and `oxm_hasmask` value, `oxm_length` is a constant. It is included only to allow software to minimally parse OXM TLVs of unknown types. (Similarly, for a given `oxm_class`, `oxm_field`, and `oxm_length`, `oxm_hasmask` is a constant.)

#### A.2.3.3 OXM classes

The match types are structured using OXM match classes. The OpenFlow specification distinguish two types of OXM match classes, ONF member classes and ONF reserved classes, differentiated by their high order bit. Classes with the high order bit set to 1 are ONF reserved classes, they are used for the OpenFlow specification itself. Classes with the high order bit set to zero are ONF member classes, they are allocated by the ONF on an as needed basis, they uniquely identify an ONF member and can be used arbitrarily by that member. Support for ONF member classes is optional.

The following OXM classes are defined:

```
/* OXM Class IDs.
 * The high order bit differentiate reserved classes from member classes.
 * Classes 0x0000 to 0x7FFF are member classes, allocated by ONF.
 * Classes 0x8000 to 0xFFFE are reserved classes, reserved for standardisation.
 */
enum ofp_oxm_class {
    OFPXMC_NXM_0          = 0x0000,    /* Backward compatibility with NXM */
    OFPXMC_NXM_1          = 0x0001,    /* Backward compatibility with NXM */
    OFPXMC_OPENFLOW_BASIC = 0x8000,    /* Basic class for OpenFlow */
    OFPXMC_EXPERIMENTER   = 0xFFFF,    /* Experimenter class */
};
```

The class `OFPXMC_OPENFLOW_BASIC` contains the basic set of OpenFlow match fields (see A.2.3.7). The optional class `OFPXMC_EXPERIMENTER` is used for experimenter matches (see A.2.3.8). Other ONF reserved classes are reserved for future uses such as modularisation of the specification. The first two ONF member classes `OFPXMC_NXM_0` and `OFPXMC_NXM_1` are reserved for backward compatibility with the Nicira Extensible Match (NXM) specification.

#### A.2.3.4 Flow Matching

A zero-length OpenFlow match (one with no OXM TLVs) matches every packet. Match fields that should be wildcarded are omitted from the OpenFlow match.

An OXM TLV places a constraint on the packets matched by the OpenFlow match:

- If `oxm_hasmask` is 0, the OXM TLV's body contains a value for the field, called `oxm_value`. The OXM TLV match matches only packets in which the corresponding field equals `oxm_value`.

| No. | '111 Patent Claim 15 | The Reference |
|---|---|---|
| 15[b] | wherein the analyzing comprises checking part of, or whole of, the payload field. | The Reference discloses wherein the analyzing comprises checking part of, or whole of, the payload field.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| 16[a] | The method according to claim 1, wherein the packet comprises distinct header and payload fields, | The Reference discloses the method according to claim 1, wherein the packet comprises distinct header and payload fields.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic comprised of packets with headers and payloads. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254 |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | • Kumar '739<br><br>Khan '478 at 7:31-49 ("TLOC: A TLOC is similar to the NEXT HOP attribute in BGP and is carried in the overlay route NLRI with a type value of 1. The actual TLOC is not carried as an immediate attribute to the prefix, but rather the System-IP of the OMP speaker originating the overlay route. Carrying the System-IP allows for the mapping between overlay routes and TLOCs irrespectively of what the actual TLOC happens to be. This is important since TLOCs can change and will change when traversing NATs, something that OMP is designed to take into consideration. This TLOC attribute points the TLOC AFI/SAFI. Within the SAFI for each TLOC, the detailed information on each specific TLOC can be found. This includes detailed information on the actual next-hop address to use, the actual TLOC. This information includes the public IP address of the TLOC and if NAT is involved, the private and non-translated TLOC-address. This separation of information allows for individual adver-tisement and invalidation of overlay routes or TLOCs with-out having to invalidate the other dependent entity.")<br><br>Wang '735 at 5:20-39 ("In one example, routers 12 at the ends of WAN link 16 comprise the optimization system 18. An RTP (Real-time Transport Protocol) trunk may be used between two adjacent optimization system devices 12 on a media path. For example, as shown in FIG. 1, the RTP trunk may be between two optimization system devices 12 connected by WAN link 16 on which bandwidth optimization is needed. The RTP trunk is preferably configured to support one of more of the following features to reduce overhead, reduce redundant cop-ies of streams, create a branch out RTP trunk, or carry addi-tional flags or markings. For example, RTP header compres-sion and session multiplexing may be used to reduce overhead. Data Redundancy Elimination (DRE) may be used with UDP-based real-time multimedia applications to reduce redundant copies of video/audio streams in a multiple point conference or live-streaming applications. In order to provide DRE, both WAN optimization devices 12 maintain a synchro-nization RTP payload cache. The RTP header may be pre-served over the trunk and the payload may be encoded by an index in a cache buffer, for example.")<br><br>Wang '735 at 11:27-49 ("Another media optimization mechanism that the system may utilize is congestion control for variable bit-rate video applications. Congestion may occur, for example, when over-subscription occurs and applications generate more traffic than a |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | network link can transport, when multiple video encod-ers send burst traffic onto the network, when there is no QoS provisioning or inappropriate QoS settings on router, or when network bandwidth changes mid-session. In these cases, the router has to drop packets if no buffer is available to store the excess traffic. Conversation video streams should not be buff-ered during congestion in order to minimize the delay and jitter. Techniques that may be used include video DPI (Deep Packet Inspection) and video specific application parsing (e.g., parse the H.264 RTP header and video payload) to extract video specification information from a flow or from multiple packets of the flow, such as priority, entropy of a packet or flow, video quality score, frame boundary, etc. Intelligent (selective) packet dropping to drop less important packets first or SVC (Scalable Video Coding) layer filtering and forwarding may also be used. Bandwidth and resource CAC may be used during session setup and mid-call (e.g., preemption, over-subscription/down speeding, resume/re-cover).")<br><br>Olofsson '254 at 7:29-43 ("In one embodiment, each service router in the path of a service chain accepts inbound traffic based on the destina-tion TLOC and VPN Label in the received packet and forwards it out the associated interface for the specific service being associated with the TLOC/Label combination. In the outbound direction, each service router must be equipped with policy describing what the next hop is for the particular destination. This allows for each service router to support multiple service chains and different policies for each direction of traffic. Since the outgoing direction is controlled by policy, this allows for great flexibility in<br>choosing the next point in the service chain based on individually defined criteria for that service chain, service, or service router.")<br><br>Kumar '739 at 1:62-2:22 ("Service chaining primarily involves the interception of traffic and steering the traffic through a series of service nodes (i.e., physical or virtual devices) that each host one or more service-functions. The traffic is intercepted through the use of a classifier function at a node (i.e., switch, router, etc.) that serves as a head-end node to the service chain. The node that executes the classifier function is sometimes referred to herein as a "classifier" or "classifier node." In general, the traffic is steered from the classifier through the service-functions using one or more Layer 2 (L2)/Layer 3 (L3) service overlays in the network. In addition, a service header is appended to the traffic for forwarding through the service chain and the service header enables the carrying of service metadata in addition to the original data/payload. |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | A service header is part of the data-plane of a service chain and includes metadata specifically formatted for con- sumption by a service-function. The metadata may include, for example, an application identifier (ID), flow or path ID, and client or user ID, network classification information used for deriving targeted service policies and profiles, common metadata related to a particular service such as finer classification that can be passed to the service-func-tions further down the service-path. In other words, service-functions benefit from metadata derived both from the network as well as the service-functions that form a given service chain. Metadata can also be passed between network nodes and be used, for example, to determine forwarding state at the end of a service chain.")<br><br>Kumar '739 at 3:47-65 ("The service nodes 35, 40, 45, 50, and 55 each host/support one or more service-functions (services) for application to the payload of traffic passing through the respective service node. More specifically, service node 35 hosts service- functions 65(1) (service-function f1 ), 65(2) (service-func-tion f2), and 65(3) (service-function f3), while service node 40 hosts service-functions 65(3) (service-function f3), 65(5)( service-function f5), 65( 6) ( service-function f6), and 65(7)(service-function f7). Service node 45 hosts service-functions 65(1) (service-function f1 ), 65(5) (service-function f5), and 65(10) (service-function f10), while service node 50 hosts service-functions 65(3) (service-function f3), 65(5)(service-function f5), and 65(10) (service-function f10). Finally, service node 55 hosts service-functions 65(2) (service-function f2) and 65(3) (service-function f3). As shown, service-functions may appear in multiple instances on dif-ferent service nodes or on the same service node. For example, service-function f3 is hosted on each of the service nodes 35, 40, 50 and 55.")<br><br>Kumar '739 at 5:34-47 ("In the example of FIG. 2, classification and mapping logic 75 selects service-functions from several different service nodes. In particular, classification and mapping logic 75 selects service-functions f1 and f2 at service node 35, ser-vice-functions f6 and f7 at service node 40, and service-function f10 at service node 45. The path for service-function chain SFC1 selected by classification and mapping logic 75 is shown in FIG. 2 by broken line 100. The classifier 30 sends traffic 90 along the path 100 using one or more L2/L3/L4 service overlays in the network. In other words, a service header is appended to the traffic 90 for forwarding through the service chain and the service header enables the carrying of service metadata in addition to the original data/payload.") |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| 16[b] | the header comprises one or more flag bits, and | The Reference discloses the header comprises one or more flag bits.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, and Balakrishnan.<br><br>Below are examples of such references.<br><br>OpenFlow at 56 |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | The `flags` field may include the follow flags:<br><br>`enum ofp_flow_mod_flags {`<br>`  OFPFF_SEND_FLOW_REM = 1 << 0,  /* Send flow removed message when flow`<br>`  * expires or is deleted. */`<br>`  OFPFF_CHECK_OVERLAP = 1 << 1,  /* Check for overlapping entries first. */`<br>`  OFPFF_RESET_COUNTS  = 1 << 2,  /* Reset flow packet and byte counts. */`<br>`  OFPFF_NO_PKT_COUNTS = 1 << 3,  /* Don't keep track of packet count. */`<br>`  OFPFF_NO_BYT_COUNTS = 1 << 4,  /* Don't keep track of byte count. */`<br>`};`<br><br>When the `OFPFF_SEND_FLOW_REM` flag is set, the switch must send a flow removed message when the flow entry expires or is deleted.<br><br>When the `OFPFF_CHECK_OVERLAP` flag is set, the switch must check that there are no conflicting entries with the same priority prior to inserting it in the flow table. If there is one, the flow mod fails and an error message is returned (see 6.4).<br><br>When the `OFPFF_NO_PKT_COUNTS` flag is set, the switch does not need to keep track of the flow packet count. When the `OFPFF_NO_BYT_COUNTS` flag is set, the switch does not need to keep track of the flow byte count. Setting those flags may decrease the processing load on some OpenFlow switches, however those counters may not be available in flow statistics and flow removed messages for this flow entry. A switch is not required to honor those flags and may keep track of a flow count and return it despite the corresponding flag being set. If a switch does not keep track of a flow count, the corresponding counter is not available and must be set to the maximum field value (see 5.8).<br><br>When a flow entry is inserted in a table, its `flags` field is set with the values from the message. When a flow entry is matched and modified (`OFPFC_MODIFY` or `OFPFC_MODIFY_STRICT` messages), the `flags` field is ignored.<br><br>The `instructions` field contains the instruction set for the flow entry when adding or modifying entries. If the instruction set is not valid or supported, the switch must generate an error (see 6.4).<br><br>Balakrishnan at [0019] ("A set of criteria corresponding to the type of objects is then created, at step 104. In various embodiments of the invention, the set of criteria corresponds to at least one packet field associated with a configuration of the network device. A packet flag refers to one of the packet fields. In various embodiments of the invention, a packet flag corre-sponds to the type of objects. For example, if the packet is a TCP packet, then the packet flag is a TCP flag. Exemplary TCP flags may be for example, syn, ack, fin, urg, psh, and rst. In an embodiment of the invention, the set of criteria include at least one 'set criterion' for selecting the packet. In another embodiment of the invention, the set of criteria include at least one 'not-set criterion' for rejecting the packet. In yet another embodiment of the invention, the set of criteria includes at least one 'set criterion' and at least one 'not-set |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | criterion' for managing the network traffic in the network device. Further, in various embodiments of the invention, if a packet flag is not present in the ACL, then it does not matter if the packet flag is present or not present in the packet.")<br><br>Balakrishnan at [0026] ("Means for providing 502 provides a plurality of objects associated with a packet of the network traffic. The objects associated with the packet may be, for example the packet fields. In an embodiment of the invention, the object associated with a TCP packet may be TCP flags. In various embodiments of the invention, means for providing 502 may be a software module.")<br><br>Balakrishnan at [0028]-[0035] ("Means for matching 506 correspondingly matches the objects of the network device with the objects associated with the packet based upon the set of criteria, as described in conjunction with FIG. 1. In various embodiments of the invention, means for matching 506 may be a software module. In an embodiment of the invention, the matching is performed by using keywords such as 'match-any' and 'match-all', programmed on an Internetworking Operation Systems (IOS) of the network device. 'Match-any' keyword transmits the packet if at least one criterion from the set of criteria is satisfied. For example, a command for TCP flag filtering, also referred to as Access Control Entry (ACE), entered on the Command Line Interface (CLI) may be,<br><br>permit tcp <src> <dst> match-any+syn+ack<br>This ACE allows the packet if at least one of syn and ack are set on the packet. In an embodiment of the inven-tion, the code for 'Match-any' keyword may be as follows:<br><br>if ((match_flags & match_mask) ' (-packet_ flags & match_mask))<br><br>'Match-all' keyword transmits the packet if each criterion from the set of criteria is satisfied. For example, an ACE, for TCP flag filtering entered on the CLI may be,<br><br>permit tcp <src> <dst> match-all+syn+ack-fin<br>This ACE allows the packet only if syn and ack are set and fin is not set on the packet. In an embodiment of the invention, the code for 'Match-any' keyword may be as follows:<br><br>if ((match_flags & match_mask)==(packet_flags |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | & match_mask))<br>In a further example, an ip ACL for TCP flag filtering entered on the CLI may be,<br><br>permit tcp any any match-all+syn+ack-fin<br><br>deny tcp any any match-any+psh-rst<br><br>permit ip any any<br><br>The first ACE allows the packet only if syn and ack are set and fin is not set on the packet. The second ACE does not allow the packet if psh is set or rst is not set on the packet. The last ACE accepts all packets.") |
| 16[c] | wherein the packet applicable criterion is that one or more of the flag bits is set. | The Reference discloses wherein the packet applicable criterion is that one or more of the flag bits is set.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, and Balakrishnan.<br><br>Below are examples of such references.<br><br>OpenFlow at 56 |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | The `flags` field may include the follow flags:<br><br>`enum ofp_flow_mod_flags {`<br>`  OFPFF_SEND_FLOW_REM = 1 << 0,  /* Send flow removed message when flow`<br>`  * expires or is deleted. */`<br>`  OFPFF_CHECK_OVERLAP = 1 << 1,  /* Check for overlapping entries first. */`<br>`  OFPFF_RESET_COUNTS  = 1 << 2,  /* Reset flow packet and byte counts. */`<br>`  OFPFF_NO_PKT_COUNTS = 1 << 3,  /* Don't keep track of packet count. */`<br>`  OFPFF_NO_BYT_COUNTS = 1 << 4,  /* Don't keep track of byte count. */`<br>`};`<br><br>When the `OFPFF_SEND_FLOW_REM` flag is set, the switch must send a flow removed message when the flow entry expires or is deleted.<br><br>When the `OFPFF_CHECK_OVERLAP` flag is set, the switch must check that there are no conflicting entries with the same priority prior to inserting it in the flow table. If there is one, the flow mod fails and an error message is returned (see 6.4).<br><br>When the `OFPFF_NO_PKT_COUNTS` flag is set, the switch does not need to keep track of the flow packet count. When the `OFPFF_NO_BYT_COUNTS` flag is set, the switch does not need to keep track of the flow byte count. Setting those flags may decrease the processing load on some OpenFlow switches, however those counters may not be available in flow statistics and flow removed messages for this flow entry. A switch is not required to honor those flags and may keep track of a flow count and return it despite the corresponding flag being set. If a switch does not keep track of a flow count, the corresponding counter is not available and must be set to the maximum field value (see 5.8).<br><br>When a flow entry is inserted in a table, its `flags` field is set with the values from the message. When a flow entry is matched and modified (`OFPFC_MODIFY` or `OFPFC_MODIFY_STRICT` messages), the `flags` field is ignored.<br><br>The `instructions` field contains the instruction set for the flow entry when adding or modifying entries. If the instruction set is not valid or supported, the switch must generate an error (see 6.4).<br><br>Balakrishnan at [0028]-[0035] ("Means for matching 506 correspondingly matches the objects of the network device with the objects associated with the packet based upon the set of criteria, as described in conjunction with FIG. 1. In various embodiments of the invention, means for matching 506 may be a software module. In an embodiment of the invention, the matching is performed by using keywords such as 'match-any' and 'match-all', programmed on an Internetworking Operation Systems (IOS) of the network device. 'Match-any' keyword transmits the packet if at least one criterion from the set of criteria is satisfied. For example, a command for TCP flag filtering, also referred to as Access Control Entry (ACE), entered on the Command Line Interface (CLI) may be, |

| No. | '111 Patent Claim 16 | The Reference |
|---|---|---|
| | | permit tcp <src> <dst> match-any+syn+ack<br>This ACE allows the packet if at least one of syn and ack are set on the packet. In an embodiment of the inven-tion, the code for 'Match-any' keyword may be as follows:<br><br>if ((match_flags & match_mask) ' (-packet_ flags & match_mask))<br><br>'Match-all' keyword transmits the packet if each criterion from the set of criteria is satisfied. For example, an ACE, for TCP flag filtering entered on the CLI may be,<br><br>permit tcp <src> <dst> match-all+syn+ack-fin<br>This ACE allows the packet only if syn and ack are set and fin is not set on the packet. In an embodiment of the invention, the code for 'Match-any' keyword may be as follows:<br><br>if ((match_flags & match_mask)==(packet_flags<br>& match_mask))<br>In a further example, an ip ACL for TCP flag filtering entered on the CLI may be,<br><br>permit tcp any any match-all+syn+ack-fin<br><br>deny tcp any any match-any+psh-rst<br><br>permit ip any any<br><br>The first ACE allows the packet only if syn and ack are set and fin is not set on the packet. The second ACE does not allow the packet if psh is set or rst is not set on the packet. The last ACE accepts all packets.") |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| 17[a] | The method according to claim 16, wherein the packet is an Transmission Control Protocol (TCP) packet, and | The Reference discloses the method according to claim 16, wherein the packet is an Transmission Control Protocol (TCP) packet.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Balakrishnan, Khan '478, Wang '735, and Olofsson '254.<br><br>Below are examples of such references.<br><br>OpenFlow at 24-25<br>**6.3 OpenFlow Channel Connections**<br>The OpenFlow channel is used to exchange OpenFlow message between an OpenFlow switch and an OpenFlow controller. A typical OpenFlow controller manages multiple OpenFlow channels, each one to a different OpenFlow switch. An OpenFlow switch may have one OpenFlow channel to a single controller, or multiple channels for reliability, each to a different controller (see 6.3.4).<br><br>An OpenFlow controller typically manages an OpenFlow switch remotely over one or more networks. The specification of the networks used for the OpenFlow channels is outside the scope of the present specification. It may be a separate dedicated network, or the OpenFlow channel may use the network managed by the OpenFlow switch (in-band controller connection). The only requirement is that it should provide TCP/IP connectivity.<br><br>The OpenFlow channel is usually instantiated as a single network connection, using TLS or plain TCP (see 6.3.3). The OpenFlow channel may be composed of multiple network connections to exploit parallelism (see 6.3.5). The OpenFlow switch always initiates a connection to an OpenFlow controller (see 6.3.1).<br>**6.3.1 Connection Setup**<br>The switch must be able to establish communication with a controller at a user-configurable (but otherwise fixed) IP address, using a user-specified port. If the switch knows the IP address of the controller, the switch initiates a standard TLS or TCP connection to the controller. Traffic to and from the OpenFlow channel is not run through the OpenFlow pipeline. Therefore, the switch must identify incoming traffic as local before checking it against the flow tables.<br><br>When an OpenFlow connection is first established, each side of the connection must immediately send an `OFPT_HELLO` message with the `version` field set to the highest OpenFlow protocol version supported by the sender. Upon receipt of this message, the recipient may calculate the OpenFlow protocol version to be used as the smaller of the version number that it sent and the one that it received.<br><br>If the negotiated version is supported by the recipient, then the connection proceeds. Otherwise, the recipient must reply with an `OFPT_ERROR` message with a `type` field of `OFPET_HELLO_FAILED`, a `code` field of `OFPHFC_COMPATIBLE`, and optionally an ASCII string explaining the situation in `data`, and then terminate the connection. |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | Balakrishnan at [0018] ("FIG. 1 is a flowchart illustrating a method for managing network traffic in a network device, in accordance with an exemplary embodiment of the invention. At step 102, a plurality of objects associated with a packet of the network traffic is provided. A packet generally refers to a unit of data, which can be of any protocol type. A packet may be a Transmission Control Protocol (TCP) packet. The objects associated with the packet may be, for example, a source port of the packet and a destination port of the packet.")<br><br>Balakrishnan at [0019] ("A set of criteria corresponding to the type of objects is then created, at step 104. In various embodiments of the invention, the set of criteria corresponds to at least one packet field associated with a configuration of the network device. A packet flag refers to one of the packet fields. In various embodiments of the invention, a packet flag corre-sponds to the type of objects. For example, if the packet is a TCP packet, then the packet flag is a TCP flag. Exemplary TCP flags may be for example, syn, ack, fin, urg, psh, and rst. In an embodiment of the invention, the set of criteria include at least one 'set criterion' for selecting the packet. In another embodiment of the invention, the set of criteria include at least one 'not-set criterion' for rejecting the packet. In yet another embodiment of the invention, the set of criteria includes at least one 'set criterion' and at least one 'not-set criterion' for managing the network traffic in the network device. Further, in various embodiments of the invention, if a packet flag is not present in the ACL, then it does not matter if the packet flag is present or not present in the packet.")<br><br>Balakrishnan at [0026] ("Means for providing 502 provides a plurality of objects associated with a packet of the network traffic. The objects associated with the packet may be, for example the packet fields. In an embodiment of the invention, the object associated with a TCP packet may be TCP flags. In various embodiments of the invention, means for providing 502 may be a software module.")<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic using a TCP protocol. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735 |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | • Olofsson '254<br><br>Khan '478 at 4:10-15 ("In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing. The OMP may listen on TCP port [17900, assigned through IRNA].")<br><br>Wang '735 at 1:41-62 ("In one embodiment, a method generally comprises receiv-ing application traffic at a network device from one or more endpoints, measuring performance of applications at the net-work device, optimizing TCP (Transmission Control Proto-col) applications and UDP (User Datagram Protocol) appli-cations based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available bandwidth in accordance with the measured perfor-mance and the policy input, and transmitting the application traffic over a wide area network.<br>In another embodiment, an apparatus generally comprises a performance manager for measuring application perfor-mance for application traffic received at the apparatus, a UDP optimizer for optimizing UDP applications, a TCP optimizer for optimizing TCP applications, a policy manager for pro-cessing policy input received at the apparatus, and a plurality of queues for queuing the application traffic received at the apparatus based on input from the performance manager and policy manager. The application traffic shares available band-width in a wide area network in accordance with the measured performance and policy input.")<br><br>Wang '735 at 2:38-62 ("The embodiments described herein provide a complete solution to effectively integrate all optimization features to minimize deployment cost and provide additional features and value to users. As described in detail below, the embodi-ments manage business critical applications, real time voice/ video, and best effort traffic automatically and effectively. The embodiments provide for efficient use of network resources for TCP traffic and UDP traffic including real-time communications and non-real-time streaming. The embodi-ments may be used, for example, to effectively optimize UDP-based real-time conversational UC&C (Unified Com-munications and Collaboration) applications. Different types of application traffic contending the WAN bandwidth are |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | managed together and a system-level and network-level architecture for a complete solution is provided. This allows for pervasive video deployment, management of application performance requirements, and effective delivery of quality of services to networked applications and users. As described below, WAN optimization system components may be embedded in a network device such as a router, therefore eliminating the need for additional network devices. The WAN optimization system components may operate at a branch office, data center, or Internet edge, thus eliminating the need for different solutions for different network loca-tions.")<br><br>Wang '735 at 6:55-7:2 ("FIG. 3 illustrates an example of the WAN optimization system 18 at one of the optimization system devices 12. The system 18 includes a performance manager (APM (applica-tion performance manager)/VQM (video/voice quality man-ager)) 30, policy manager 32, TCP-based application optimi-zation module 34, UDP-based application optimization module 36, and scheduling and queuing module 38. The node 12 receives input (signaling) to the optimization system 18 from other nodes (e.g., routers with optimization system installed, endpoints 10). The node 12 also sends optimization system output to other routers and endpoints 10. Policy input is provided to the node 12 from an external policy source, as described below. The input may be received from down- stream and upstream devices and the output may be transmit-ted to downstream and upstream devices.")<br><br>Wang '735 at 8:24-35 ("The TCP-based application optimization module 34 is configured to optimize all TCP-based applications (also referred to herein as TCP applications), including streaming video and audio. In one embodiment, the optimization module 34 is a Cisco WAAS (Wide Area Application Services) module. The optimization module 34 may request additional bandwidth from the scheduler 38 to meet application performance base- line requirements, for example. The scheduler 38 may proac-tively throttle the TCP output to yield more bandwidth to higher priority UDP-based video applications that are man-aged by the media optimization module 36, based on policy and performance baselines.")<br><br>Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated overlay control plane protocol over underlying network infrastructure 108. In one embodi-ment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.") |
| 17[b] | wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof. | The Reference discloses wherein the one or more flag bits comprises comprise a SYN flag bit, an ACK flag bit, a FIN flag bit, a RST flag bit, or any combination thereof.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and Balakrishnan.<br><br>Below are examples of such references.<br><br>Balakrishnan at [0019] ("A set of criteria corresponding to the type of objects is then created, at step 104. In various embodiments of the invention, the set of criteria corresponds to at least one packet field associated with a configuration of the network device. A packet flag refers to one of the packet fields. In various embodiments of the invention, a packet flag corre-sponds to the type of objects. For example, if the packet is a TCP packet, then the packet flag is a TCP flag. Exemplary TCP flags may be for example, syn, ack, fin, urg, psh, and rst. In an embodiment of the invention, the set of criteria include at least one 'set criterion' for selecting the packet. In another embodiment of the invention, the set of criteria include at least one 'not-set criterion' for rejecting the packet. In yet another embodiment of the invention, the set of criteria includes at least one 'set criterion' and at least one 'not-set |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | criterion' for managing the network traffic in the network device. Further, in various embodiments of the invention, if a packet flag is not present in the ACL, then it does not matter if the packet flag is present or not present in the packet.")<br><br>Balakrishnan at [0026] ("Means for providing 502 provides a plurality of objects associated with a packet of the network traffic. The objects associated with the packet may be, for example the packet fields. In an embodiment of the invention, the object associated with a TCP packet may be TCP flags. In various embodiments of the invention, means for providing 502 may be a software module.")<br><br>Balakrishnan at [0028]-[0035] ("Means for matching 506 correspondingly matches the objects of the network device with the objects associated with the packet based upon the set of criteria, as described in conjunction with FIG. 1. In various embodiments of the invention, means for matching 506 may be a software module. In an embodiment of the invention, the matching is performed by using keywords such as 'match-any' and 'match-all', programmed on an Internetworking Operation Systems (IOS) of the network device. 'Match-any' keyword transmits the packet if at least one criterion from the set of criteria is satisfied. For example, a command for TCP flag filtering, also referred to as Access Control Entry (ACE), entered on the Command Line Interface (CLI) may be,<br><br>permit tcp <src> <dst> match-any+syn+ack<br>This ACE allows the packet if at least one of syn and ack are set on the packet. In an embodiment of the inven-tion, the code for 'Match-any' keyword may be as follows:<br><br>if ((match_flags & match_mask) ' (-packet_ flags & match_mask))<br><br>'Match-all' keyword transmits the packet if each criterion from the set of criteria is satisfied. For example, an ACE, for TCP flag filtering entered on the CLI may be,<br><br>permit tcp <src> <dst> match-all+syn+ack-fin<br>This ACE allows the packet only if syn and ack are set and fin is not set on the packet. In an embodiment of the invention, the code for 'Match-any' keyword may be as follows:<br><br>if ((match_flags & match_mask)==(packet_flags |

| No. | '111 Patent Claim 17 | The Reference |
|---|---|---|
| | | & match_mask))<br>In a further example, an ip ACL for TCP flag filtering entered on the CLI may be,<br><br>permit tcp any any match-all+syn+ack-fin<br><br>deny tcp any any match-any+psh-rst<br><br>permit ip any any<br><br>The first ACE allows the packet only if syn and ack are set and fin is not set on the packet. The second ACE does not allow the packet if psh is set or rst is not set on the packet. The last ACE accepts all packets.") |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| 18[a] | The method according to claim 1, wherein the packet comprises distinct header and payload fields, | The Reference discloses the method according to claim 1, wherein the packet comprises distinct header and payload fields.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic comprised of packets with headers and payloads. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254 |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| | | • Kumar '739<br><br>Khan '478 at 7:31-49 ("TLOC: A TLOC is similar to the NEXT HOP attribute in BGP and is carried in the overlay route NLRI with a type value of 1. The actual TLOC is not carried as an immediate attribute to the prefix, but rather the System-IP of the OMP speaker originating the overlay route. Carrying the System-IP allows for the mapping between overlay routes and TLOCs irrespectively of what the actual TLOC happens to be. This is important since TLOCs can change and will change when traversing NATs, something that OMP is designed to take into consideration. This TLOC attribute points the TLOC AFI/SAFI. Within the SAFI for each TLOC, the detailed information on each specific TLOC can be found. This includes detailed information on the actual next-hop address to use, the actual TLOC. This information includes the public IP address of the TLOC and if NAT is involved, the private and non-translated TLOC-address. This separation of information allows for individual adver-tisement and invalidation of overlay routes or TLOCs with-out having to invalidate the other dependent entity.")<br><br>Wang '735 at 5:20-39 ("In one example, routers 12 at the ends of WAN link 16 comprise the optimization system 18. An RTP (Real-time Transport Protocol) trunk may be used between two adjacent optimization system devices 12 on a media path. For example, as shown in FIG. 1, the RTP trunk may be between two optimization system devices 12 connected by WAN link 16 on which bandwidth optimization is needed. The RTP trunk is preferably configured to support one of more of the following features to reduce overhead, reduce redundant cop-ies of streams, create a branch out RTP trunk, or carry addi-tional flags or markings. For example, RTP header compres-sion and session multiplexing may be used to reduce overhead. Data Redundancy Elimination (DRE) may be used with UDP-based real-time multimedia applications to reduce redundant copies of video/audio streams in a multiple point conference or live-streaming applications. In order to provide DRE, both WAN optimization devices 12 maintain a synchro-nization RTP payload cache. The RTP header may be pre-served over the trunk and the payload may be encoded by an index in a cache buffer, for example.")<br><br>Wang '735 at 11:27-49 ("Another media optimization mechanism that the system may utilize is congestion control for variable bit-rate video applications. Congestion may occur, for example, when over-subscription occurs and applications generate more traffic than a |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| | | network link can transport, when multiple video encod-ers send burst traffic onto the network, when there is no QoS provisioning or inappropriate QoS settings on router, or when network bandwidth changes mid-session. In these cases, the router has to drop packets if no buffer is available to store the excess traffic. Conversation video streams should not be buff-ered during congestion in order to minimize the delay and jitter. Techniques that may be used include video DPI (Deep Packet Inspection) and video specific application parsing (e.g., parse the H.264 RTP header and video payload) to extract video specification information from a flow or from multiple packets of the flow, such as priority, entropy of a packet or flow, video quality score, frame boundary, etc. Intelligent (selective) packet dropping to drop less important packets first or SVC (Scalable Video Coding) layer filtering and forwarding may also be used. Bandwidth and resource CAC may be used during session setup and mid-call (e.g., preemption, over-subscription/down speeding, resume/re-cover).")<br><br>Olofsson '254 at 7:29-43 ("In one embodiment, each service router in the path of a service chain accepts inbound traffic based on the destina-tion TLOC and VPN Label in the received packet and forwards it out the associated interface for the specific service being associated with the TLOC/Label combination. In the outbound direction, each service router must be equipped with policy describing what the next hop is for the particular destination. This allows for each service router to support multiple service chains and different policies for each direction of traffic. Since the outgoing direction is controlled by policy, this allows for great flexibility in<br>choosing the next point in the service chain based on individually defined criteria for that service chain, service, or service router.")<br><br>Kumar '739 at 1:62-2:22 ("Service chaining primarily involves the interception of traffic and steering the traffic through a series of service nodes (i.e., physical or virtual devices) that each host one or more service-functions. The traffic is intercepted through the use of a classifier function at a node (i.e., switch, router, etc.) that serves as a head-end node to the service chain. The node that executes the classifier function is sometimes referred to herein as a "classifier" or "classifier node." In general, the traffic is steered from the classifier through the service-functions using one or more Layer 2 (L2)/Layer 3 (L3) service overlays in the network. In addition, a service header is appended to the traffic for forwarding through the service chain and the service header enables the carrying of service metadata in addition to the original data/payload. |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| | | A service header is part of the data-plane of a service chain and includes metadata specifically formatted for con- sumption by a service-function. The metadata may include, for example, an application identifier (ID), flow or path ID, and client or user ID, network classification information used for deriving targeted service policies and profiles, common metadata related to a particular service such as finer classification that can be passed to the service-func-tions further down the service-path. In other words, service-functions benefit from metadata derived both from the network as well as the service-functions that form a given service chain. Metadata can also be passed between network nodes and be used, for example, to determine forwarding state at the end of a service chain.")<br><br>Kumar '739 at 3:47-65 ("The service nodes 35, 40, 45, 50, and 55 each host/support one or more service-functions (services) for application to the payload of traffic passing through the respective service node. More specifically, service node 35 hosts service- functions 65(1) (service-function f1 ), 65(2) (service-func-tion f2), and 65(3) (service-function f3), while service node 40 hosts service-functions 65(3) (service-function f3), 65(5)( service-function f5), 65( 6) ( service-function f6), and 65(7)(service-function f7). Service node 45 hosts service-functions 65(1) (service-function f1 ), 65(5) (service-function f5), and 65(10) (service-function f10), while service node 50 hosts service-functions 65(3) (service-function f3), 65(5)(service-function f5), and 65(10) (service-function f10). Finally, service node 55 hosts service-functions 65(2) (service-function f2) and 65(3) (service-function f3). As shown, service-functions may appear in multiple instances on dif-ferent service nodes or on the same service node. For example, service-function f3 is hosted on each of the service nodes 35, 40, 50 and 55.")<br><br>Kumar '739 at 5:34-47 ("In the example of FIG. 2, classification and mapping logic 75 selects service-functions from several different service nodes. In particular, classification and mapping logic 75 selects service-functions f1 and f2 at service node 35, ser-vice-functions f6 and f7 at service node 40, and service-function f10 at service node 45. The path for service-function chain SFC1 selected by classification and mapping logic 75 is shown in FIG. 2 by broken line 100. The classifier 30 sends traffic 90 along the path 100 using one or more L2/L3/L4 service overlays in the network. In other words, a service header is appended to the traffic 90 for forwarding through the service chain and the service header enables the carrying of service metadata in addition to the original data/payload.") |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| 18[b] | the header comprises at least the first and second entities addresses in the packet network, and | The Reference discloses the header comprises at least the first and second entities addresses in the packet network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>OpenFlow at 24-25<br>**6.3 OpenFlow Channel Connections**<br>The OpenFlow channel is used to exchange OpenFlow message between an OpenFlow switch and an OpenFlow controller. A typical OpenFlow controller manages multiple OpenFlow channels, each one to a different OpenFlow switch. An OpenFlow switch may have one OpenFlow channel to a single controller, or multiple channels for reliability, each to a different controller (see 6.3.4).<br><br>An OpenFlow controller typically manages an OpenFlow switch remotely over one or more networks. The specification of the networks used for the OpenFlow channels is outside the scope of the present specification. It may be a separate dedicated network, or the OpenFlow channel may use the network managed by the OpenFlow switch (in-band controller connection). The only requirement is that it should provide TCP/IP connectivity.<br><br>The OpenFlow channel is usually instantiated as a single network connection, using TLS or plain TCP (see 6.3.3). The OpenFlow channel may be composed of multiple network connections to exploit |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|

parallelism (see 6.3.5). The OpenFlow switch always initiates a connection to an OpenFlow controller (see 6.3.1).

### 6.3.1 Connection Setup

The switch must be able to establish communication with a controller at a user-configurable (but otherwise fixed) IP address, using a user-specified port. If the switch knows the IP address of the controller, the switch initiates a standard TLS or TCP connection to the controller. Traffic to and from the OpenFlow channel is not run through the OpenFlow pipeline. Therefore, the switch must identify incoming traffic as local before checking it against the flow tables.

When an OpenFlow connection is first established, each side of the connection must immediately send an `OFPT_HELLO` message with the `version` field set to the highest OpenFlow protocol version supported by the sender. Upon receipt of this message, the recipient may calculate the OpenFlow protocol version to be used as the smaller of the version number that it sent and the one that it received.

If the negotiated version is supported by the recipient, then the connection proceeds. Otherwise, the recipient must reply with an `OFPT_ERROR` message with a `type` field of `OFPET_HELLO_FAILED`, a `code` field of `OFPHFC_COMPATIBLE`, and optionally an ASCII string explaining the situation in `data`, and then terminate the connection.

Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic comprised of packets with source, destination, and port addresses. Some examples of Cisco's patents for that technology that are relevant to this limitation include:

- Khan '478
- Olofsson '254
- Kumar '739

Khan '478 at 7:31-49 ("TLOC: A TLOC is similar to the NEXT HOP attribute in BGP and is carried in the overlay route NLRI with a type value of 1. The actual TLOC is not carried as an immediate attribute to the prefix, but rather the System-IP of the OMP speaker originating the overlay route. Carrying the System-IP allows for the mapping between overlay routes and TLOCs irrespectively of what the actual TLOC happens to be. This is important since TLOCs can change and will change when traversing NATs, something that OMP is designed to take into consideration. This TLOC attribute points the TLOC AFI/SAFI. Within the SAFI for each TLOC, the detailed information on each specific TLOC

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| | | can be found. This includes detailed information on the actual next-hop address to use, the actual TLOC. This information includes the public IP address of the TLOC and if NAT is involved, the private and non-translated TLOC-address. This separation of information allows for individual adver-tisement and invalidation of overlay routes or TLOCs with-out having to invalidate the other dependent entity.")<br><br>Olofsson '254 at 2:45-63 ("In one embodiment, in order for the overlay control plane protocol to provide a functional architecture, it distributes overlay routes that are learned from each location where an overlay network element is present, together with external addresses used as next-hop addresses for the overlay routes. The external addresses may be assigned to the physical interfaces of the overlay network elements that attach to the underlying network 108. In one embodiment, the overlay routes may only be accessed through the overlay network 100 and the next-hop addresses can only be reached through the underlying network 108. Together, the overlay routes and next-hop addresses provide for a complete and func-tional overlay architecture, as will be explained. As far as the underlying network 108 is concerned, the only element used to forward traffic between the sites is the next-hop address. The underlying network 108 does not know about any other routes, addresses or labels that may be used for providing a functional network infrastructure within the overlay network 100 itself.")<br><br>Olofsson '254 at 5:1-7 ("Block 404: Every Edge-router receives an outbound policy (for traffic towards the Internet) stipulating that all the traffic matching the routes received from Hub3, will be encapsulated in a packet with a Service-label of 1, matching the Firewall Service, and a next-hop address of Hub2. This will ensure all traffic destined for the Internet is using the tunnel from the Edge-router to Hub2")<br><br>Kumar '739 at 5:61-6:6 ("In certain circumstances, service-functions may change the flow specification (e.g., the 5-tuple comprises the source IP address, destination IP address, source port number, destination port number and the protocol in use) of pro- cessed packets. That is, traffic may be received at a service-function with a certain 5 tuple, but, after processing at the service-function, the traffic will include a different 5 tuple. When the flow specification of traffic is changed, the traffic may need to be processed by different service-functions than those identified in the initial service-function chain. Service-functions that are |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| | | capable of changing the flow specification of traffic are sometimes referred to herein as modifying service-functions.") |
| 18[c] | wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses. | The Reference discloses wherein the packet-applicable criterion is that the first entity address, the second entity address, or both match a predetermined address or addresses.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>OpenFlow at 24-25<br>**6.3 OpenFlow Channel Connections**<br>The OpenFlow channel is used to exchange OpenFlow message between an OpenFlow switch and an OpenFlow controller. A typical OpenFlow controller manages multiple OpenFlow channels, each one to a different OpenFlow switch. An OpenFlow switch may have one OpenFlow channel to a single controller, or multiple channels for reliability, each to a different controller (see 6.3.4).<br><br>An OpenFlow controller typically manages an OpenFlow switch remotely over one or more networks. The specification of the networks used for the OpenFlow channels is outside the scope of the present specification. It may be a separate dedicated network, or the OpenFlow channel may use the network managed by the OpenFlow switch (in-band controller connection). The only requirement is that it should provide TCP/IP connectivity.<br><br>The OpenFlow channel is usually instantiated as a single network connection, using TLS or plain TCP (see 6.3.3). The OpenFlow channel may be composed of multiple network connections to exploit |

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|

parallelism (see 6.3.5). The OpenFlow switch always initiates a connection to an OpenFlow controller (see 6.3.1).

### 6.3.1 Connection Setup

The switch must be able to establish communication with a controller at a user-configurable (but otherwise fixed) IP address, using a user-specified port. If the switch knows the IP address of the controller, the switch initiates a standard TLS or TCP connection to the controller. Traffic to and from the OpenFlow channel is not run through the OpenFlow pipeline. Therefore, the switch must identify incoming traffic as local before checking it against the flow tables.

When an OpenFlow connection is first established, each side of the connection must immediately send an `OFPT_HELLO` message with the `version` field set to the highest OpenFlow protocol version supported by the sender. Upon receipt of this message, the recipient may calculate the OpenFlow protocol version to be used as the smaller of the version number that it sent and the one that it received.

If the negotiated version is supported by the recipient, then the connection proceeds. Otherwise, the recipient must reply with an `OFPT_ERROR` message with a `type` field of `OFPET_HELLO_FAILED`, a `code` field of `OFPHFC_COMPATIBLE`, and optionally an ASCII string explaining the situation in `data`, and then terminate the connection.

Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic comprised of packets with source, destination, and port addresses. Some examples of Cisco's patents for that technology that are relevant to this limitation include:

- Khan '478
- Olofsson '254
- Kumar '739

Khan '478 at 7:31-49 ("TLOC: A TLOC is similar to the NEXT HOP attribute in BGP and is carried in the overlay route NLRI with a type value of 1. The actual TLOC is not carried as an immediate attribute to the prefix, but rather the System-IP of the OMP speaker originating the overlay route. Carrying the System-IP allows for the mapping between overlay routes and TLOCs irrespectively of what the actual TLOC happens to be. This is important since TLOCs can change and will change when traversing NATs, something that OMP is designed to take into consideration. This TLOC attribute points the TLOC AFI/SAFI. Within the SAFI for each TLOC, the detailed information on each specific TLOC can be found. This includes detailed information on the actual next-hop address to use, the

| No. | '111 Patent Claim 18 | The Reference |
|---|---|---|
| | | actual TLOC. This information includes the public IP address of the TLOC and if NAT is involved, the private and non-translated TLOC-address. This separation of information allows for individual adver-tisement and invalidation of overlay routes or TLOCs with-out having to invalidate the other dependent entity.")<br><br>Olofsson '254 at 2:45-63 ("In one embodiment, in order for the overlay control plane protocol to provide a functional architecture, it distributes overlay routes that are learned from each location where an overlay network element is present, together with external addresses used as next-hop addresses for the overlay routes. The external addresses may be assigned to the physical interfaces of the overlay network elements that attach to the underlying network 108. In one embodiment, the overlay routes may only be accessed through the overlay network 100 and the next-hop addresses can only be reached through the underlying network 108. Together, the overlay routes and next-hop addresses provide for a complete and func-tional overlay architecture, as will be explained. As far as the underlying network 108 is concerned, the only element used to forward traffic between the sites is the next-hop address. The underlying network 108 does not know about any other routes, addresses or labels that may be used for providing a functional network infrastructure within the overlay network 100 itself.")<br><br>Olofsson '254 at 5:1-7 ("Block 404: Every Edge-router receives an outbound policy (for traffic towards the Internet) stipulating that all the traffic matching the routes received from Hub3, will be encapsulated in a packet with a Service-label of 1, matching the Firewall Service, and a next-hop address of Hub2. This will ensure all traffic destined for the Internet is using the tunnel from the Edge-router to Hub2")<br><br>Kumar '739 at 5:61-6:6 ("In certain circumstances, service-functions may change the flow specification (e.g., the 5-tuple comprises the source IP address, destination IP address, source port number, destination port number and the protocol in use) of pro- cessed packets. That is, traffic may be received at a service-function with a certain 5 tuple, but, after processing at the service-function, the traffic will include a different 5 tuple. When the flow specification of traffic is changed, the traffic may need to be processed by different service-functions than those identified in the initial service-function chain. Service-functions that are capable of changing the flow specification of traffic are sometimes referred to herein as modifying service-functions.") |

| No. | '111 Patent Claim 19 | The Reference |
|---|---|---|
| 19 | The method according to claim 18, wherein the addresses are Internet Protocol (IP) addresses. | The Reference discloses the method according to claim 18, wherein the addresses are Internet Protocol (IP) addresses.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, and Balakrishnan,.<br><br>Below are examples of such references.<br><br>OpenFlow at 24-25<br>**6.3 OpenFlow Channel Connections**<br>The OpenFlow channel is used to exchange OpenFlow message between an OpenFlow switch and an OpenFlow controller. A typical OpenFlow controller manages multiple OpenFlow channels, each one to a different OpenFlow switch. An OpenFlow switch may have one OpenFlow channel to a single controller, or multiple channels for reliability, each to a different controller (see 6.3.4).<br><br>An OpenFlow controller typically manages an OpenFlow switch remotely over one or more networks. The specification of the networks used for the OpenFlow channels is outside the scope of the present specification. It may be a separate dedicated network, or the OpenFlow channel may use the network managed by the OpenFlow switch (in-band controller connection). The only requirement is that it should provide TCP/IP connectivity.<br><br>The OpenFlow channel is usually instantiated as a single network connection, using TLS or plain TCP (see 6.3.3). The OpenFlow channel may be composed of multiple network connections to exploit |

| No. | '111 Patent Claim 19 | The Reference |
|---|---|---|
| | | parallelism (see 6.3.5). The OpenFlow switch always initiates a connection to an OpenFlow controller (see 6.3.1).<br><br>**6.3.1 Connection Setup**<br><br>The switch must be able to establish communication with a controller at a user-configurable (but otherwise fixed) IP address, using a user-specified port. If the switch knows the IP address of the controller, the switch initiates a standard TLS or TCP connection to the controller. Traffic to and from the OpenFlow channel is not run through the OpenFlow pipeline. Therefore, the switch must identify incoming traffic as local before checking it against the flow tables.<br><br>When an OpenFlow connection is first established, each side of the connection must immediately send an `OFPT_HELLO` message with the `version` field set to the highest OpenFlow protocol version supported by the sender. Upon receipt of this message, the recipient may calculate the OpenFlow protocol version to be used as the smaller of the version number that it sent and the one that it received.<br><br>If the negotiated version is supported by the recipient, then the connection proceeds. Otherwise, the recipient must reply with an `OFPT_ERROR` message with a `type` field of `OFPET_HELLO_FAILED`, a `code` field of `OFPHFC_COMPATIBLE`, and optionally an ASCII string explaining the situation in `data`, and then terminate the connection.<br><br>Balakrishnan at [0018] ("FIG. 1 is a flowchart illustrating a method for managing network traffic in a network device, in accordance with an exemplary embodiment of the invention. At step 102, a plurality of objects associated with a packet of the network traffic is provided. A packet generally refers to a unit of data, which can be of any protocol type. A packet may be a Transmission Control Protocol (TCP) packet. The objects associated with the packet may be, for example, a source port of the packet and a destination port of the packet.") |

| No. | '111 Patent Claim 20 | The Reference |
|---|---|---|
| 20[a] | The method according to claim 1, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and TCP sequence mask fields, and | The Reference discloses the method according to claim 1, wherein the packet is an Transmission Control Protocol (TCP) packet that comprises source and destination TCP ports, a TCP sequence number, and TCP sequence mask fields.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and Balakrishnan.<br><br>Below are examples of such references.<br><br>Balakrishnan at [0018] ("FIG. 1 is a flowchart illustrating a method for managing network traffic in a network device, in accordance with an exemplary embodiment of the invention. At step 102, a plurality of objects associated with a packet of the network traffic is provided. A packet generally refers to a unit of data, which can be of any protocol type. A packet may be a Transmission Control Protocol (TCP) packet. The objects associated with the packet may be, for example, a source port of the packet and a destination port of the packet.")<br><br>Balakrishnan at [0019] ("A set of criteria corresponding to the type of objects is then created, at step 104. In various embodiments of the invention, the set of criteria corresponds to at least one packet field associated with a configuration of the network device. A packet flag refers to one of the packet fields. In various embodiments of the invention, a packet flag corre-sponds to the type of objects. For example, if the packet is a TCP packet, then the packet flag is a TCP flag. Exemplary TCP flags may be for example, syn, ack, fin, urg, psh, and rst. In an embodiment of the invention, the set of criteria include at least one 'set criterion' for selecting the packet. In another embodiment of the invention, the set of criteria include at least one 'not-set criterion' for rejecting the packet. In yet another embodiment of the invention, the set of criteria includes at least one 'set criterion' and at least one 'not-set criterion' for managing the network traffic in the network device. Further, in various embodiments of the invention, |

| No. | '111 Patent Claim 20 | The Reference |
|---|---|---|
| | | if a packet flag is not present in the ACL, then it does not matter if the packet flag is present or not present in the packet.") |
| 20[b] | wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any combination thereof, matches a predetermined value or values. | The Reference discloses wherein the packet-applicable criterion is that the source TCP port, the destination TCP port, the TCP sequence number, the TCP sequence mask, or any combination thereof, matches a predetermined value or values.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, and Balakrishnan.<br><br>Below are examples of such references.<br><br>OpenFlow at 24-25<br>**6.3 OpenFlow Channel Connections**<br>The OpenFlow channel is used to exchange OpenFlow message between an OpenFlow switch and an OpenFlow controller. A typical OpenFlow controller manages multiple OpenFlow channels, each one to a different OpenFlow switch. An OpenFlow switch may have one OpenFlow channel to a single controller, or multiple channels for reliability, each to a different controller (see 6.3.4).<br><br>An OpenFlow controller typically manages an OpenFlow switch remotely over one or more networks. The specification of the networks used for the OpenFlow channels is outside the scope of the present specification. It may be a separate dedicated network, or the OpenFlow channel may use the network managed by the OpenFlow switch (in-band controller connection). The only requirement is that it should provide TCP/IP connectivity.<br><br>The OpenFlow channel is usually instantiated as a single network connection, using TLS or plain TCP (see 6.3.3). The OpenFlow channel may be composed of multiple network connections to exploit |

| No. | '111 Patent Claim 20 | The Reference |
|---|---|---|
| | | parallelism (see 6.3.5). The OpenFlow switch always initiates a connection to an OpenFlow controller (see 6.3.1).<br><br>**6.3.1 Connection Setup**<br><br>The switch must be able to establish communication with a controller at a user-configurable (but otherwise fixed) IP address, using a user-specified port. If the switch knows the IP address of the controller, the switch initiates a standard TLS or TCP connection to the controller. Traffic to and from the OpenFlow channel is not run through the OpenFlow pipeline. Therefore, the switch must identify incoming traffic as local before checking it against the flow tables.<br><br>When an OpenFlow connection is first established, each side of the connection must immediately send an `OFPT_HELLO` message with the `version` field set to the highest OpenFlow protocol version supported by the sender. Upon receipt of this message, the recipient may calculate the OpenFlow protocol version to be used as the smaller of the version number that it sent and the one that it received.<br><br>If the negotiated version is supported by the recipient, then the connection proceeds. Otherwise, the recipient must reply with an `OFPT_ERROR` message with a `type` field of `OFPET_HELLO_FAILED`, a `code` field of `OFPHFC_COMPATIBLE`, and optionally an ASCII string explaining the situation in `data`, and then terminate the connection.<br><br>Balakrishnan at [0019] ("A set of criteria corresponding to the type of objects is then created, at step 104. In various embodiments of the invention, the set of criteria corresponds to at least one packet field associated with a configuration of the network device. A packet flag refers to one of the packet fields. In various embodiments of the invention, a packet flag corre-sponds to the type of objects. For example, if the packet is a TCP packet, then the packet flag is a TCP flag. Exemplary TCP flags may be for example, syn, ack, fin, urg, psh, and rst. In an embodiment of the invention, the set of criteria include at least one 'set criterion' for selecting the packet. In another embodiment of the invention, the set of criteria include at least one 'not-set criterion' for rejecting the packet. In yet another embodiment of the invention, the set of criteria includes at least one 'set criterion' and at least one 'not-set criterion' for managing the network traffic in the network device. Further, in various embodiments of the invention, if a packet flag is not present in the ACL, then it does not matter if the packet flag is present or not present in the packet.") |

| No. | '111 Patent Claim 21 | The Reference |
|---|---|---|
| 21 | The method according to claim 1, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet, Metropolitan Area Network (MAN), Internet service Provider (IsP) backbone datacenter network, or inter - datacenter network. | The Reference discloses the method according to claim 1, wherein the packet network comprises a Wide Area Network (WAN), Local Area Network (LAN), the Internet, Metropolitan Area Network (MAN), Internet service Provider (IsP) backbone datacenter network, or inter - datacenter network.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Khan '478, Wang '735, and Olofsson '254.<br><br>Below are examples of such references.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic over a variety of networks. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br><br>Khan '478 at 3:15-22 ("In use, the overlay domain (OD) 100 may rely on a transport network 120 to provide network transport func-tionality, as will be described later. The transport network 120 may include any wide area network (WAN) and in some embodiments may include the Internet, other public WAN, a Metro Ethernet or MPLS. Typically, the transport network 120 may include circuits and networks provided by third parties such as carriers, and service provides (SPs).")<br><br>Khan '478 at 10:20-38 ("The hardware also typically receives a number of inputs and outputs for communicating information externally. For interface with a user or operator, the hardware |

| No. | '111 Patent Claim 21 | The Reference |
|---|---|---|
| | | may include one or more user input output devices 806 ( e.g., a keyboard, mouse, etc.) and a display 808. For additional storage, the hardware 800 may also include one or more mass storage devices 810, e.g., a Universal Serial Bus (USB) or other removable disk drive, a hard disk drive, a Direct Access Storage Device (DASD), an optical drive (e.g. a Compact Disk (CD) drive, a Digital Versatile Disk (DVD) drive, etc.) and/or a USB drive, among others. Furthermore, the hard- ware may include an interface with one or more networks 812 ( e.g., a local area network (LAN), a wide area network (WAN), a wireless network, and/or the Internet among others) to permit the communication of information with other computers coupled to the networks. It should be appreciated that the hardware typically includes suitable analog and/or digital interfaces between the processor 812 and each of the components, as is well known in the art.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.")<br><br>Wang '735 at 1:41-62 ("In one embodiment, a method generally comprises receiv-ing application traffic at a network device from one or more endpoints, measuring performance of applications at the net-work device, optimizing TCP (Transmission Control Proto-col) applications and UDP (User Datagram Protocol) appli-cations based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available bandwidth in accordance with the measured perfor-mance and the policy input, and transmitting the application traffic over a wide area network.<br>In another embodiment, an apparatus generally comprises a performance manager for measuring application perfor-mance for application traffic received at the apparatus, a UDP optimizer for optimizing UDP applications, a TCP optimizer for optimizing TCP applications, a policy manager for pro-cessing policy input received at the apparatus, and a |

| No. | '111 Patent Claim 21 | The Reference |
|---|---|---|
| | | plurality of queues for queuing the application traffic received at the apparatus based on input from the performance manager and policy manager. The application traffic shares available band-width in a wide area network in accordance with the measured performance and policy input.")<br><br>Wang '735 at 2:12-36 ("Multi-tiered services operate over WANs (Wide Area Net-works) and many applications share bandwidth on the net-work. These include, for example, unmanaged applications ( e.g., Internet streaming, Internet VoIP (Voice over Internet Protocol)), video applications ( e.g., IP video conference, sur-veillance, video telephony, HD (High Definition) video con-ference), voice applications (e.g., IP Telephony), and data applications ( e.g., application sharing, Internet, messaging, e-mail). When deploying video applications on wide area network links that are usually oversubscribed and overloaded, business critical applications compete against lower priority traffic and video applications for bandwidth. Because both TCP (Transmission Control Protocol) and UDP (User Data- gram Protocol) run over the same network, an increase in UDP traffic in conventional systems impacts the performance of applications using TCP. Conventional systems used to deploy video applications do not monitor and optimize the performance of each application. Business productivity applications may be impacted by non-critical traffic without any visibility and protection. Also, unused bandwidth in one queue may not be used by other queues automatically and effectively, which results in inefficiency of a dedicated queue. Conventional systems do not automatically react in real-time to the bandwidth demand needed to maintain satisfied user experience.")<br><br>Wang '735 at 3:41-64 ("The embodiments described herein operate in the context of a data communication system including multiple network elements. Referring now to the drawings, and first to FIG. 1, an example of a network in which embodiments described herein may be implemented is shown. The communication system comprises a plurality of endpoints 10 in communica-tion through a plurality of network devices ( e.g., routers) 12 and over networks 14. The communication system may include any number of networks ( e.g., local area network, metropolitan area network, wide area network, enterprise network, Internet, intranet, radio access network, public switched network, or any other network or combination of networks). The flow path between the endpoints 10 may include any number or type of intermediate nodes ( e.g., rout-ers, switches, gateways, management stations, appliances, or |

| No. | '111 Patent Claim 21 | The Reference |
|---|---|---|
| | | other network devices), which facilitate passage of data between the endpoints. Also, there may be any number of endpoints 10. The endpoints 10 may be located at a branch office, for example, and in communication with an ISR (Inte-grated Services Router) 12 connected to a WAN access link 13. The ISRs may be in communication with ASRs (Aggre-gated Services Router) 12 operating at the network edge, for example. The routers 12 communicate over a wide area net-work.")<br><br>Wang '735 at 9:29-49 ("FIG. 4 is a flowchart illustrating an overview of a process for WAN optimization, in accordance with one embodiment. At step 40, the network device (e.g., router 12) receives poli-cies and input from other WAN optimization system nodes and endpoints 10. The network device 12 identifies applica-tion traffic received from a plurality of endpoints 10 and associated with a plurality of applications (step 42). As described above, various methods may be used to recognize applications and identify flows. Flow information may be stored at the device 12. The performance manager 30 moni-tors the input traffic and measures the application perfor-mance at the network device 12 (step 44). As previously described, the application performance may be compared against baseline performance requirements. The router opti-mizes TCP and UDP traffic based on the measured perfor-mance and policy input (step 46). The traffic is queued at the network device (step 48) and transmitted from the network device 12 over the wide area network (step 50). The applica-tion traffic (for UDP and TCP applications) share available WAN bandwidth in accordance with the measured perfor-mance and policy input.<br><br>Olofsson '254 at 8:60-9:11 ("The hardware also typically receives a number of inputs and outputs for communicating information externally. For interface with a user or operator, the hardware may include one or more user input output devices 606 ( e.g., a keyboard, mouse, etc.) and a display 608. For additional storage, the hardware 600 may also include one or more mass storage devices 610, e.g., a Universal Serial Bus (USB) or other removable disk drive, a hard disk drive, a Direct Access Storage Device (DASD), an optical drive (e.g. a Compact Disk (CD) drive, a Digital Versatile Disk (DVD) drive, etc.) and/or a USB drive, among others. Furthermore, the hard-ware may include an interface with one or more networks 612 ( e.g., a local area network (LAN), a wide area network (WAN), a wireless network, and/or the Internet among others) to permit the communication of information with other computers coupled to the networks. It should be appreciated that the hardware typically |

| No. | '111 Patent Claim 21 | The Reference |
|---|---|---|
| | | includes suitable analog and/or digital interfaces between the processor 612 and each of the components, as is well known in the art.") |

| No. | '111 Patent Claim 22 | The Reference |
|---|---|---|
| 22 | The method according to claim 1, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.<br><br>. | The Reference discloses the method according to claim 1, wherein the first entity is a server device and the second entity is a client device, or wherein the first entity is a client device and the second entity is a server device.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 23 | The Reference |
|---|---|---|
| 23[a] | The method according to claim 22, wherein the server device comprises a web server, and | The Reference discloses the method according to claim 22, wherein the server device comprises a web server.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of |

| No. | '111 Patent Claim 23 | The Reference |
|---|---|---|
| | | the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |
| 23[b] | wherein the client device comprises a smartphone, a tablet computer, a personal computer, a laptop computer, or a wearable computing device. | The Reference discloses wherein the client device comprises a smartphone, a tablet computer, a personal computer, a laptop computer, or a wearable computing device.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 24 | The Reference |
|---|---|---|
| 24 | The method according to claim 22, wherein the communication between the network node and the controller is based on, or uses, a standard protocol. | The Reference discloses the method according to claim 22, wherein the communication between the network node and the controller is based on, or uses, a standard protocol.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 24 | The Reference |
|---|---|---|
| | | |

| No. | '111 Patent Claim 27 | The Reference |
|---|---|---|
| 27 | The method according to claim 1, wherein the network node comprises a router, a switch, or a bridge. | The Reference discloses the method according to claim 1, wherein the network node comprises a router, a switch, or a bridge.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and OpenFlow.<br><br>Below are examples of such references.<br><br>OpenFlow at 6-7 |

| No. | '111 Patent Claim 27 | The Reference |
|---|---|---|

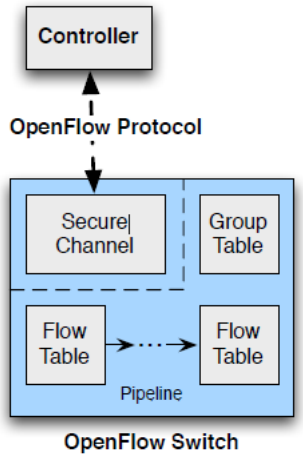



Figure 1: Main components of an OpenFlow switch.

## 2 Switch Components

An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The switch communicates with the controller and the controller manages the switch via the OpenFlow protocol.

Using the OpenFlow protocol, the controller can add, update, and delete *flow entries* in flow tables, both reactively (in response to packets) and proactively. Each flow table in the switch contains a set of flow entries; each flow entry consists of *match fields*, *counters*, and a set of *instructions* to apply to matching packets (see 5.2).

Matching starts at the first flow table and may continue to additional flow tables (see 5.1). Flow entries match packets in priority order, with the first matching entry in each table being used (see 5.3). If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on configuration of the table-miss flow entry: for example, the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table (see 5.4).

Instructions associated with each flow entry either contain actions or modify pipeline processing (see 5.9). Actions included in instructions describe packet forwarding, packet modification and group table

| No. | '111 Patent Claim 27 | The Reference |
|---|---|---|
| | | processing. Pipeline processing instructions allow packets to be sent to subsequent tables for further processing and allow information, in the form of metadata, to be communicated between tables. Table pipeline processing stops when the instruction set associated with a matching flow entry does not specify a next table; at this point the packet is usually modified and forwarded (see 5.10).<br><br>Flow entries may forward to a *port*. This is usually a physical port, but it may also be a logical port defined by the switch or a reserved port defined by this specification (see 4.1). Reserved ports may specify generic forwarding actions such as sending to the controller, flooding, or forwarding using non-OpenFlow methods, such as "normal" switch processing (see 4.5), while switch-defined logical ports may specify link aggregation groups, tunnels or loopback interfaces (see 4.4).<br><br>Actions associated with flow entries may also direct packets to a group, which specifies additional processing (see 5.6). Groups represent sets of actions for flooding, as well as more complex forwarding semantics (e.g. multipath, fast reroute, and link aggregation). As a general layer of indirection, groups also enable multiple flow entries to forward to a single identifier (e.g. IP forwarding to a common next hop). This abstraction allows common output actions across flow entries to be changed efficiently.<br><br>The group table contains group entries; each group entry contains a list of *action buckets* with specific semantics dependent on group type (see 5.6.1). The actions in one or more action buckets are applied to packets sent to the group.<br><br>Switch designers are free to implement the internals in any way convenient, provided that correct match and instruction semantics are preserved. For example, while a flow entry may use an all group to forward to multiple ports, a switch designer may choose to implement this as a single bitmask within the hardware forwarding table. Another example is matching; the pipeline exposed by an OpenFlow switch may be physically implemented with a different number of hardware tables. |

| No. | '111 Patent Claim 28 | The Reference |
|---|---|---|
| 28 | The method according to claim 1, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet. | The Reference discloses the method according to claim 1, wherein the packet network is an Internet Protocol (IP) network, and the packet is an IP packet.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, |

| No. | '111 Patent Claim 28 | The Reference |
|---|---|---|
| | | Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, and Balakrishnan.<br><br>Below are examples of such references.<br><br>Balakrishnan at [0018] ("FIG. 1 is a flowchart illustrating a method for managing network traffic in a network device, in accordance with an exemplary embodiment of the invention. At step 102, a plurality of objects associated with a packet of the network traffic is provided. A packet generally refers to a unit of data, which can be of any protocol type. A packet may be a Transmission Control Protocol (TCP) packet. The objects associated with the packet may be, for example, a source port of the packet and a destination port of the packet.") |

| No. | '111 Patent Claim 29 | The Reference |
|---|---|---|
| 29 | The method according to claim 28, wherein the packet network is an Transmission Control Protocol (TCP) network, and the packet is an TCP packet. | The Reference discloses the method according to claim 28, wherein the packet network is an Transmission Control Protocol (TCP) network, and the packet is an TCP packet.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Balakrishnan, Khan '478, Wang '735, and Olofsson '254.<br><br>Below are examples of such references.<br><br>Balakrishnan at [0018] ("FIG. 1 is a flowchart illustrating a method for managing network traffic in a network device, in accordance with an exemplary embodiment of the invention. At step 102, a plurality of objects associated with a packet of the network traffic is provided. A packet generally refers to a unit of data, which can be of any protocol type. A packet may |

| No. | '111 Patent Claim 29 | The Reference |
|---|---|---|
| | | be a Transmission Control Protocol (TCP) packet. The objects associated with the packet may be, for example, a source port of the packet and a destination port of the packet.")<br><br>Balakrishnan at [0019] ("A set of criteria corresponding to the type of objects is then created, at step 104. In various embodiments of the invention, the set of criteria corresponds to at least one packet field associated with a configuration of the network device. A packet flag refers to one of the packet fields. In various embodiments of the invention, a packet flag corre-sponds to the type of objects. For example, if the packet is a TCP packet, then the packet flag is a TCP flag. Exemplary TCP flags may be for example, syn, ack, fin, urg, psh, and rst. In an embodiment of the invention, the set of criteria include at least one 'set criterion' for selecting the packet. In another embodiment of the invention, the set of criteria include at least one 'not-set criterion' for rejecting the packet. In yet another embodiment of the invention, the set of criteria includes at least one 'set criterion' and at least one 'not-set criterion' for managing the network traffic in the network device. Further, in various embodiments of the invention, if a packet flag is not present in the ACL, then it does not matter if the packet flag is present or not present in the packet.")<br><br>Balakrishnan at [0026] ("Means for providing 502 provides a plurality of objects associated with a packet of the network traffic. The objects associated with the packet may be, for example the packet fields. In an embodiment of the invention, the object associated with a TCP packet may be TCP flags. In various embodiments of the invention, means for providing 502 may be a software module.")<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic using a TCP protocol. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254 |

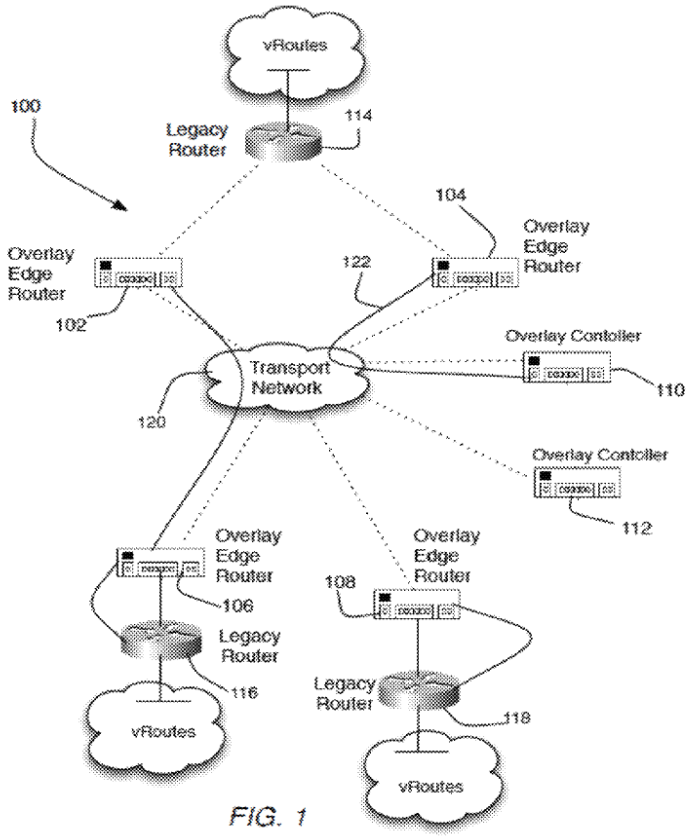
| No. | '111 Patent Claim 29 | The Reference |
|---|---|---|
| | | Khan '478 at 4:10-15 ("In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing. The OMP may listen on TCP port [17900, assigned through IRNA].")<br><br>Wang '735 at 1:41-62 ("In one embodiment, a method generally comprises receiv-ing application traffic at a network device from one or more endpoints, measuring performance of applications at the net-work device, optimizing TCP (Transmission Control Proto-col) applications and UDP (User Datagram Protocol) appli-cations based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available bandwidth in accordance with the measured perfor-mance and the policy input, and transmitting the application traffic over a wide area network.<br>In another embodiment, an apparatus generally comprises a performance manager for measuring application perfor-mance for application traffic received at the apparatus, a UDP optimizer for optimizing UDP applications, a TCP optimizer for optimizing TCP applications, a policy manager for pro-cessing policy input received at the apparatus, and a plurality of queues for queuing the application traffic received at the apparatus based on input from the performance manager and policy manager. The application traffic shares available band-width in a wide area network in accordance with the measured performance and policy input.")<br><br>Wang '735 at 2:38-62 ("The embodiments described herein provide a complete solution to effectively integrate all optimization features to minimize deployment cost and provide additional features and value to users. As described in detail below, the embodi-ments manage business critical applications, real time voice/ video, and best effort traffic automatically and effectively. The embodiments provide for efficient use of network resources for TCP traffic and UDP traffic including real-time communications and non-real-time streaming. The embodi-ments may be used, for example, to effectively optimize UDP-based real-time conversational UC&C (Unified Com-munications and Collaboration) applications. Different types of application traffic contending the WAN bandwidth are managed together and a system-level and network-level architecture for a complete solution is provided. This allows for pervasive video deployment, management of application |

| No. | '111 Patent Claim 29 | The Reference |
|---|---|---|
| | | performance requirements, and effective delivery of quality of services to networked applications and users. As described below, WAN optimization system components may be embedded in a network device such as a router, therefore eliminating the need for additional network devices. The WAN optimization system components may operate at a branch office, data center, or Internet edge, thus eliminating the need for different solutions for different network loca-tions.")<br><br>Wang '735 at 6:55-7:2 ("FIG. 3 illustrates an example of the WAN optimization system 18 at one of the optimization system devices 12. The system 18 includes a performance manager (APM (applica-tion performance manager)/VQM (video/voice quality man-ager)) 30, policy manager 32, TCP-based application optimi-zation module 34, UDP-based application optimization module 36, and scheduling and queuing module 38. The node 12 receives input (signaling) to the optimization system 18 from other nodes (e.g., routers with optimization system installed, endpoints 10). The node 12 also sends optimization system output to other routers and endpoints 10. Policy input is provided to the node 12 from an external policy source, as described below. The input may be received from down- stream and upstream devices and the output may be transmit-ted to downstream and upstream devices.")<br><br>Wang '735 at 8:24-35 ("The TCP-based application optimization module 34 is configured to optimize all TCP-based applications (also referred to herein as TCP applications), including streaming video and audio. In one embodiment, the optimization module 34 is a Cisco WAAS (Wide Area Application Services) module. The optimization module 34 may request additional bandwidth from the scheduler 38 to meet application performance base- line requirements, for example. The scheduler 38 may proac-tively throttle the TCP output to yield more bandwidth to higher priority UDP-based video applications that are man-aged by the media optimization module 36, based on policy and performance baselines.")<br><br>Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network |

| No. | '111 Patent Claim 29 | The Reference |
|---|---|---|
| | | secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated overlay control plane protocol over underlying network infrastructure 108. In one embodi-ment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.") |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| 30[a] | The method according to claim 1, further comprising: receiving, by the network node from the first entity over the packet network, one or more additional packets; | The Reference discloses the method according to claim 1, further comprising: receiving, by the network node from the first entity over the packet network, one or more additional packets.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic flows of multiple packets. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254 |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | • Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.")<br><br>Khan '478 at Figure 1 |

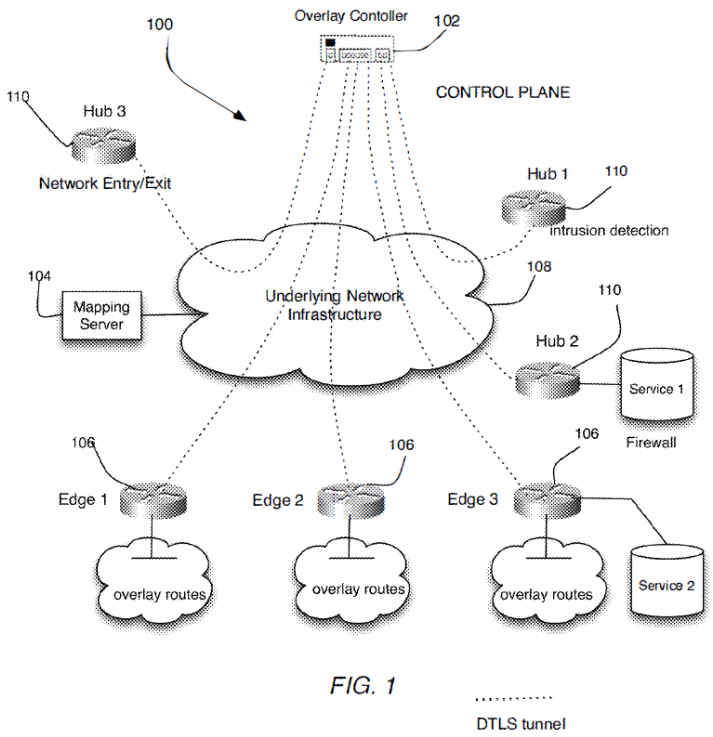
| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by reference numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.")<br><br>Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as |

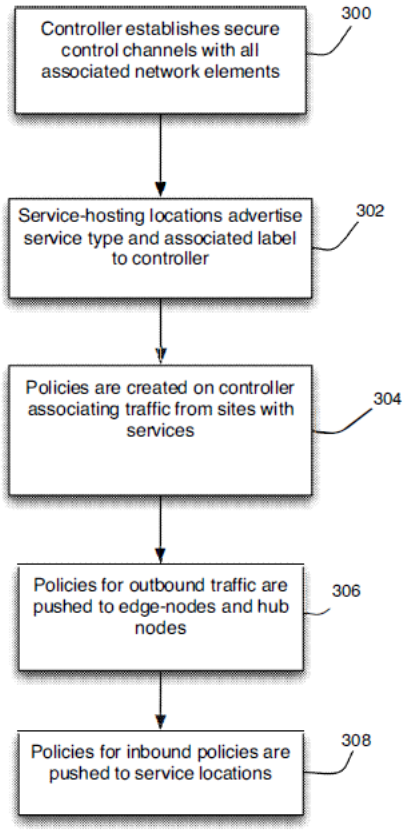
| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing.<br>The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.")<br><br>Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI).<br>In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.") |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.")<br><br>Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the applicationflows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br><br>Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 |

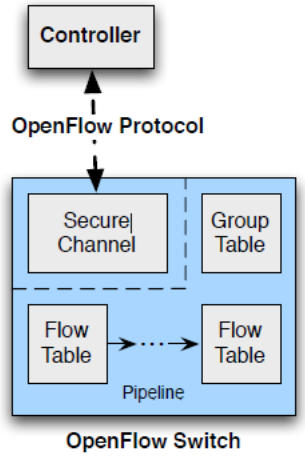
| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | related to the exact nature of the service chain that is being constructed. Two examples are provided below.")<br><br>Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.)<br><br>Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service-functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.") |

| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | | Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |
| 30[b] | checking, by the network node, if any one of the one or more additional | The Reference discloses checking, by the network node, if any one of the one or more additional packets satisfies the criterion.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was |

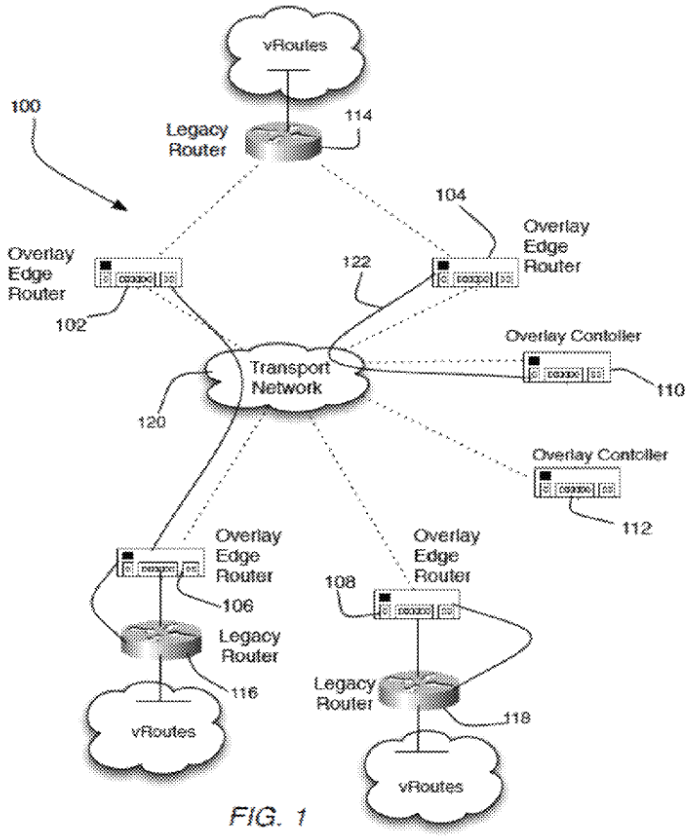
| No. | '111 Patent Claim 30 | The Reference |
|---|---|---|
| | packets satisfies the criterion; | known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |
| 30[c] | responsive to an additional packet not satisfying the criterion, sending, by the network node over the packet network, the additional packet to the second entity; and | The Reference discloses responsive to an additional packet not satisfying the criterion, sending, by the network node over the packet network, the additional packet to the second entity.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |
| 30[d] | responsive to the additional packet satisfying the criterion, sending the additional packet, by the network node over the packet network, in response to the instruction. | The Reference discloses responsive to the additional packet satisfying the criterion, sending the additional packet, by the network node over the packet network, in response to the instruction.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, and Uchida. |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| 31[a] | The method according to claim 1, wherein the packet network is a Software Defined Network (SDN), | The Reference discloses the method according to claim 1, wherein the packet network is a Software Defined Network (SDN).<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, OpenFlow, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>OpenFlow at 6-7<br>  <br><br>Figure 1: Main components of an OpenFlow switch. |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | ## 2 Switch Components<br><br>An OpenFlow Switch consists of one or more *flow tables* and a *group table*, which perform packet lookups and forwarding, and an *OpenFlow channel* to an external controller (Figure 1). The switch communicates with the controller and the controller manages the switch via the OpenFlow protocol.<br><br>Using the OpenFlow protocol, the controller can add, update, and delete *flow entries* in flow tables, both reactively (in response to packets) and proactively. Each flow table in the switch contains a set of flow entries; each flow entry consists of *match fields*, *counters*, and a set of *instructions* to apply to matching packets (see 5.2).<br><br>Matching starts at the first flow table and may continue to additional flow tables (see 5.1). Flow entries match packets in priority order, with the first matching entry in each table being used (see 5.3). If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on configuration of the table-miss flow entry: for example, the packet may be forwarded to the controller over the OpenFlow channel, dropped, or may continue to the next flow table (see 5.4).<br><br>Instructions associated with each flow entry either contain actions or modify pipeline processing (see 5.9). Actions included in instructions describe packet forwarding, packet modification and group table |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | processing. Pipeline processing instructions allow packets to be sent to subsequent tables for further processing and allow information, in the form of metadata, to be communicated between tables. Table pipeline processing stops when the instruction set associated with a matching flow entry does not specify a next table; at this point the packet is usually modified and forwarded (see 5.10).<br><br>Flow entries may forward to a *port*. This is usually a physical port, but it may also be a logical port defined by the switch or a reserved port defined by this specification (see 4.1). Reserved ports may specify generic forwarding actions such as sending to the controller, flooding, or forwarding using non-OpenFlow methods, such as "normal" switch processing (see 4.5), while switch-defined logical ports may specify link aggregation groups, tunnels or loopback interfaces (see 4.4).<br><br>Actions associated with flow entries may also direct packets to a group, which specifies additional processing (see 5.6). Groups represent sets of actions for flooding, as well as more complex forwarding semantics (e.g. multipath, fast reroute, and link aggregation). As a general layer of indirection, groups also enable multiple flow entries to forward to a single identifier (e.g. IP forwarding to a common next hop). This abstraction allows common output actions across flow entries to be changed efficiently.<br><br>The group table contains group entries; each group entry contains a list of *action buckets* with specific semantics dependent on group type (see 5.6.1). The actions in one or more action buckets are applied to packets sent to the group.<br><br>Switch designers are free to implement the internals in any way convenient, provided that correct match and instruction semantics are preserved. For example, while a flow entry may use an all group to forward to multiple ports, a switch designer may choose to implement this as a single bitmask within the hardware forwarding table. Another example is matching; the pipeline exposed by an OpenFlow switch may be physically implemented with a different number of hardware tables.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic using an overlay controller. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers |

| No. | ’111 Patent Claim 31 | The Reference |
|---|---|---|
| | | based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.”)<br><br>Khan ’478 at Figure 1 |

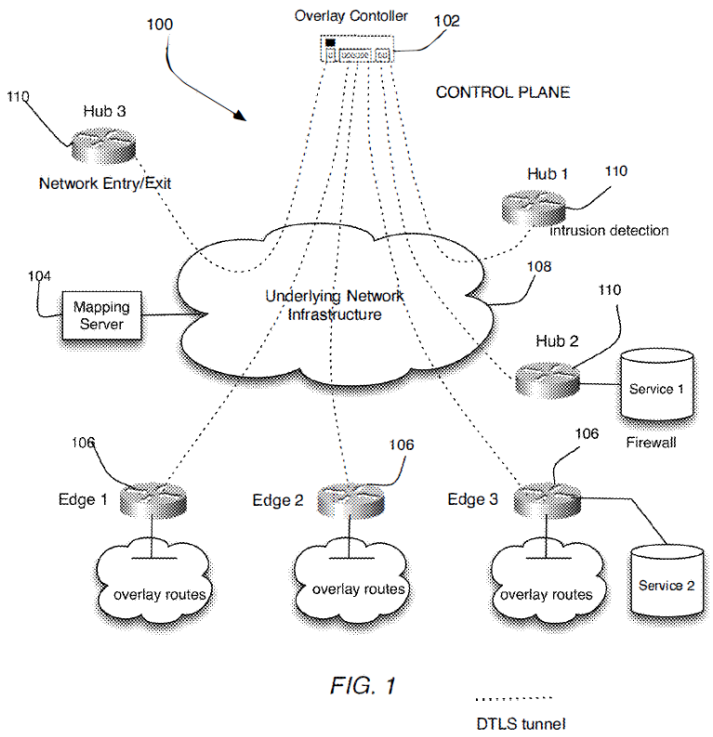
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by reference numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.")<br><br>Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as |

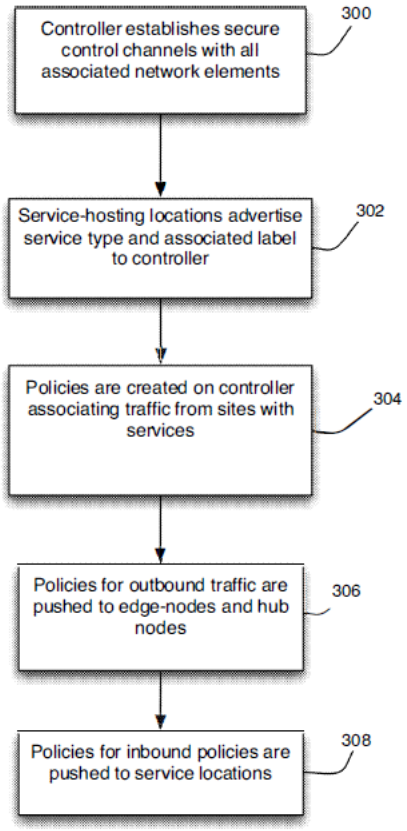
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing. The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.")<br><br>Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI).<br>In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.") |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.")<br><br>Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the application flows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br><br>Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 |

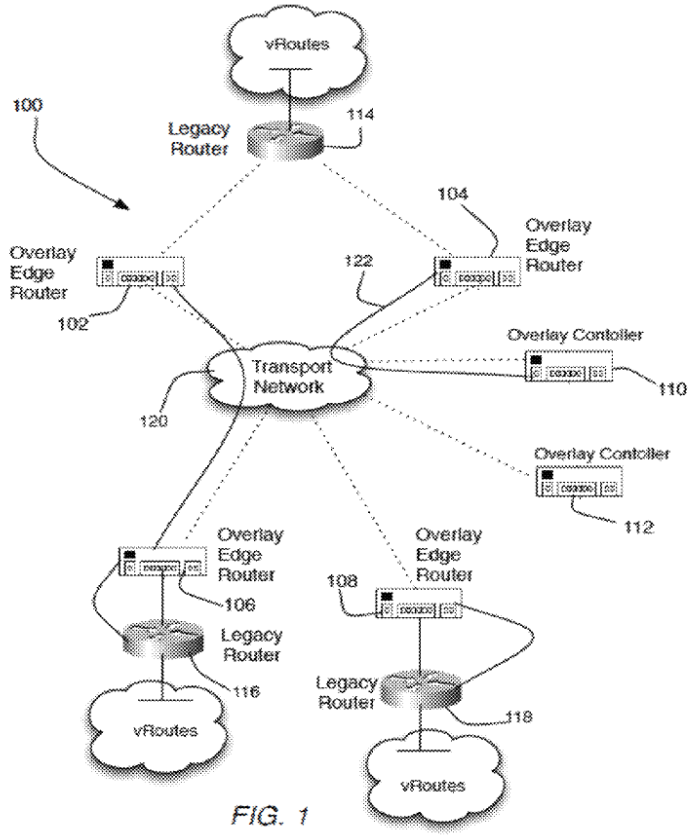
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | related to the exact nature of the service chain that is being constructed. Two examples are provided below.")<br><br>Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.)<br><br>Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service-functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.") |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| 31[b] | the packet is routed as part of a data plane and | The Reference discloses the packet is routed as part of a data plane.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic using an overlay controller. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.") |

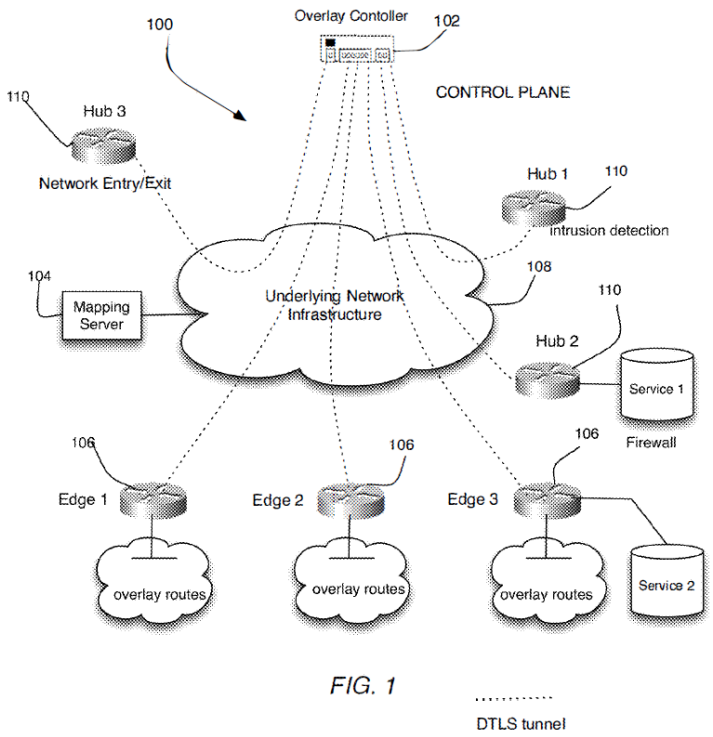
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Khan '478 at Figure 1<br>  <br><br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by reference numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.") |

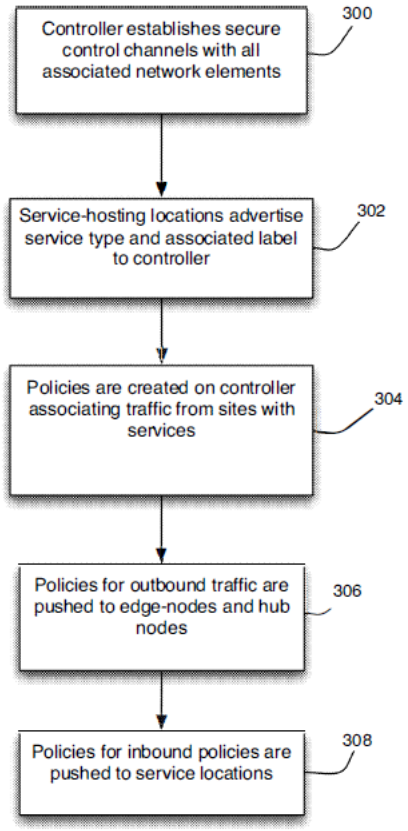
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing.<br>The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.")<br><br>Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI). |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.")<br><br>Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.") |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the applicationflows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br><br>Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are |

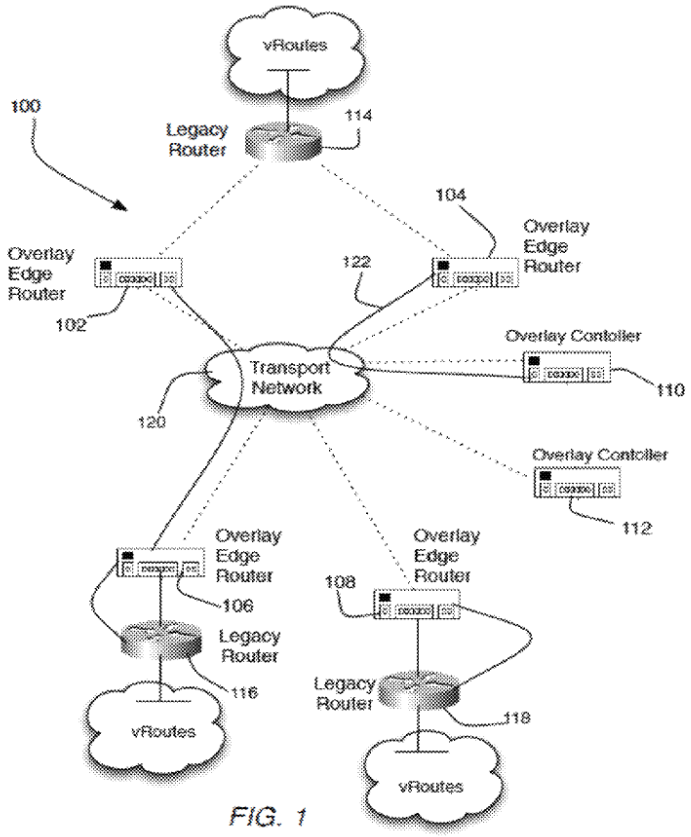
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | related to the exact nature of the service chain that is being constructed. Two examples are provided below.")<br><br>Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.)<br><br>Kumar '739 at 2:9-22 ("A service header is part of the data-plane of a service chain and includes metadata specifically formatted for consumption by a service-function. The metadata may include, for example, an application identifier (ID), flow or path ID, and client or user ID, network classification information used for deriving targeted service policies and profiles, common metadata related to a particular service such as finer classification that can be passed to the service-functions further down the service-path. In other words, service functions benefit from metadata derived both from the network as well as the service-functions that form a given service chain. Metadata can also be passed between network nodes and be used, for example, to determine forwarding state at the end of a service chain.")<br><br>Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service- |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.")<br><br>Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |
| 31[c] | the network node communication with the controller serves as a control plane. | The Reference discloses the network node communication with the controller serves as a control plane.<br><br>To the extent that the Reference alone does not anticipate or render this claim obvious, this claim, including this element would have been obvious to one skilled in the art, as it was known. For example, this claim, including this element, would have been obvious in light of the disclosures of the Reference in combination with the knowledge of a person or ordinary skill in the art and/or any of the following references: Kempf, Swenson, Chandrasekaran, Lin '400, Shieh '088, Cisco IWAN System, VMware NSX System, Chua '877, Chua '151, Copeland, Uchida, Khan '478, Wang '735, Olofsson '254, and Kumar '739.<br><br>Below are examples of such references.<br><br>Cisco also innovated, patented, or otherwise acquired various features of SD-WAN *before* Orckit's '111 patent, including use of a network node for routing network traffic using an overlay controller. Some examples of Cisco's patents for that technology that are relevant to this limitation include:<br>• Khan '478<br>• Wang '735<br>• Olofsson '254<br>• Kumar '739<br><br>Khan '478 at Abstract ("A method for creating a secure network is provided. The method comprises establishing an overlay domain to control routing between overlay edge routers |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | based on an underly-ing transport network, wherein said establishing comprises running an overlay management protocol to exchange infor-mation within the overlay domain; in accordance with the overlay management protocol defining service routes that exist exclusively within the overlay domain wherein each overlay route includes information on at least service avail-ability within the overlay domain; and selectively using the service routes to control routing between the overlay edge routers; wherein the said routing is through the underlying transport network in a manner in which said overlay routes is shared with the overlay edge routers but not with the underlying transport network via the overlay management protocol.")<br><br>Khan '478 at Figure 1 |

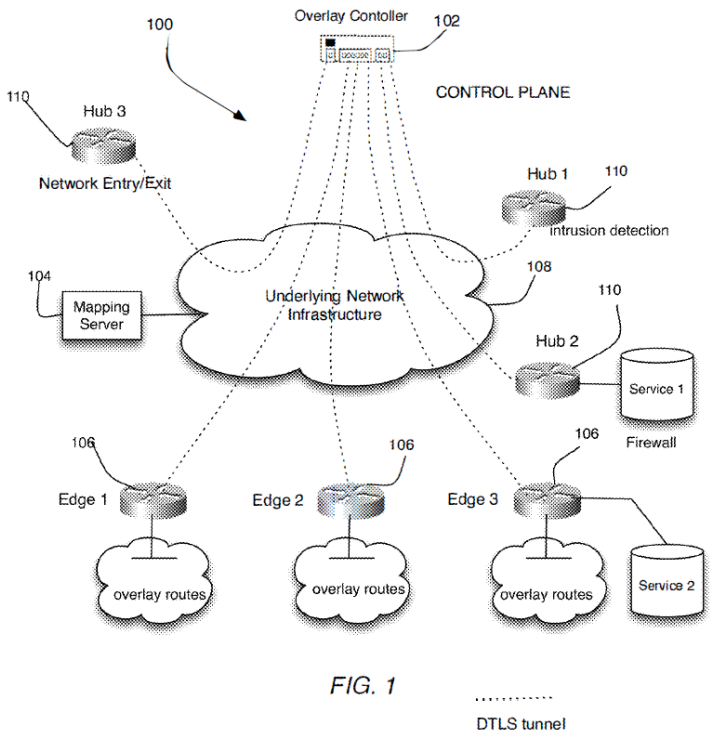
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>FIG. 1<br><br>Khan '478 at 3:1-7 ("The overlay domain (OD) 100 further comprises at least one overlay controller (OC). In FIG. 1 two overlay control-lers are shown and are indicated by reference numerals 110, and 112, respectively. As with the case of the number of the overlay edge routers (OERs), it is to be understood that the overlay domain (OD) 100 may include more or less overlay controllers than the illustrated number.")<br><br>Khan '478 at 3:49-57 ("Referring to FIG. 1, reference numeral 122 shows an example of a control channel that was established as a DTLS tunnel between the overlay edge router (OER) 104 and the overlay controller (OC) 110 via the transport network 120 as |

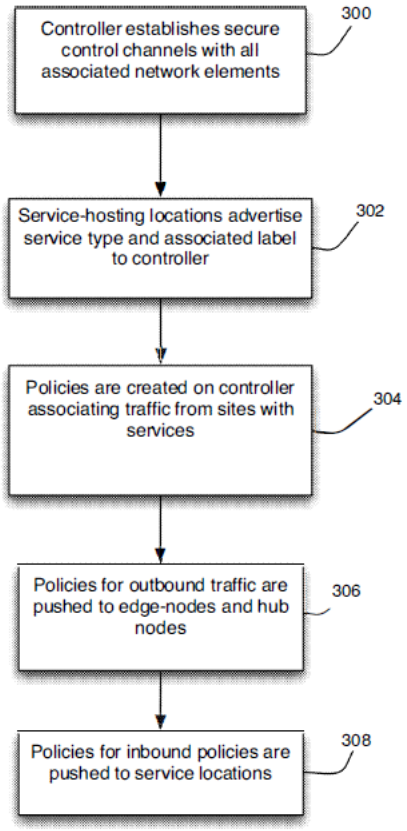
| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | a result of the bring up procedure. In one embodiment, the plurality of secure communications channels established between each overlay edge router (OER) and an assigned overlay controller (OC) together define an overlay control plane (OCP).")<br><br>Khan '478 at 4:1-17 ("In one embodiment, communications between an overlay edge router (OER) and an overlay controller (OC) may be facilitated by the use of the overlay protocol (OMP). The OMP may be used to exchange routing, policy, security, and management information between an overlay controller (OC) and an overlay edge router (OER).<br>In one embodiment, the OMP may be used to advertise routing information within the overlay domain (OD) 100, as will be described.<br>In one embodiment, the OMP uses TCP as its transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing. The OMP may listen on TCP port [17900, assigned through IRNA].<br>The OMP may be configured to handle overlay routes and transport locators (TLOCs ).")<br><br>Khan '478 at 4:47-60 ("Since the OMP operates in an overlay networking envi-ronment, the notion of routing peers is different from a traditional environment. From a logical point of view, the overlay environment consists of a central controller and a number of edge-devices. Each edge-device advertises the imported overlay routes to the central controller and the central controller, based on policy-decisions, further distrib-ute the overlay routing information to other edge-devices in the network. Edge-devices are not configured to advertise routing information to each other using the OMP. The OMP-peering sessions between overlay controller (OC) and each overlay edge router (OER) are used exclusively for the exchange of control plane traffic, whereas the overlay data plane (ODP) channels are used for data traffic.")<br><br>Khan '478 at 6:15-24 ("In one embodiment, service routes represent services connected to an overlay edge router (OER). The service routes may be advertised by the overlay edge routers (OERs) within the overlay domain (OD) 100 to the overlay control-lers (OCs) using service address family Network Layer Reachability Information (NLRI).<br>In one embodiment the OMP may be configured to redistribute the following types of routes automatically it learns either locally or from its routing peers: connected, static, OSPF intra area routes, and OSPF inter area routes.") |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Khan '478 at 9:53-10:4 ("In another embodiment, the OMP may be uses to perform a method for routing. This method is illustrated in the flowchart of FIG. 7. Referring to FIG. 7, the method may include the following processing blocks:<br>Block 700: provide an overlay network comprising at least one overlay controller; and a plurality of overlay edge routers communicatively coupled to the at least one overlay controller; wherein the overlay network is associated with an underlying transport network;<br>Block 702: collect by the overlay controller, routing information comprising at least one of authentication infor-mation, service information, encryption information, policy information, and access control information; wherein said routing information is carried by an overlay management protocol; and<br>Block 704: orchestrate by the overlay controller, routing through the underlying transport network based on the routing information; wherein said routing information is not exposed to elements of the underlying transport network.")<br><br>Wang '735 at Abstract ("In one embodiment, a method includes receiving application traffic at a network device from one or more endpoints, mea-suring performance of applications at the network device, optimizing TCP (Transmission Control Protocol) applica-tions and UDP (User Datagram Protocol) applications based on the measured performance and policy input received at the network device, queuing the application traffic at the network device such that the application traffic shares available band-width in accordance with the measured performance and the policy input, and transmitting the application traffic over a wide area network. An apparatus is also disclosed.")<br><br>Wang '735 at 7:14-38 ("The performance manager 30 may receive input from an application recognition mechanism (not shown). Application recognition features such as Cisco NBAR2 (Network Based Application Recognition 2) and MSP (Media Service Proxy) may be used to recognize networked applications. SIP/H.323/ RTSP signaling protocols may also provide a means to rec-ognize a media flow. Other input such as Cisco FnF (Flexible NetFlow) and RSVPimetadata signaling protocols may be used to aid in application recognition. An application ID or CAC (Call Admission Control) ID that is carried by RSVP/ metadata may provide additional information about the flow. The metadata may also provide, for example, |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | bandwidth res-ervation (admitted or un-admitted status), application user ID, codec type (e.g., H.264AVC, H.264 SVC, H.263, MPEG-2, etc.), maximum bandwidth (TIAS (Transport Independent Application Specific)), and minimum admitted bandwidth (for H.264: profile and level, RTP protocol and restrictions), and endpoint device capabilities (rate-adaption, Cisco Flux version supported by endpoint). The media stream may also be identified, for example, using a form of DPI (Deep Packet Inspection) or configured IP 5-tuples defining the stream. Flow information may be stored in a flow/metadata database (not shown). The flow/metadata database may be distributed to other nodes 12 incorporating the WAN optimization sys-tem.")<br><br>Wang '735 at 7:55-8:12 ("The policy manager 32 receives input from a policy server configured to receive policy information from a network administrator, for example. The policies are set up to manage application performance and resource allocations. For example, location service and service announcements may be provided for local endpoints. Policy is set up based on SLA, target performance, bit-rate, etc. Priorities are set up to meet business needs ( e.g., HD used for business is more important than regular desktop HD phone calls). The policy server may include an external network policy manager that allows thenetwork administrator to specify application classes, performance baselines per class, bandwidth usage rules, and per user SLO/SLA, etc. The policy is provisioned on all networkdevices 12 that incorporate the WAN optimization system 18 and may be implemented by a network management system (NMS), for example.<br><br>The policy manager 32 includes a network policy enforcement engine for processing policy input received at the network device 12 and managing the application delivery and performance assurance. The engine uses bandwidth pools and bandwidth usage rules defined by the policy manager to man-age and provide feedback to the other components of the optimization system 18. As shown in FIG. 3, the policy manager 32 provides input to the optimization modules 34, 36, and scheduler 38.")<br><br>Wang '735 at 10:53-11:10 ("In one embodiment, CAC (Call Admission Control) and flow policing is used to optimize media applications. CAC and flow policing may be used for on control path or when no explicit control path is involved. For on control path the control |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | message/protocol is terminated or handled by the WAN optimization system 18. In one example, RSVP (Re-source Reservation Protocol), which is used to reserve resources across the network, is used as the control protocol. The system checks the bandwidth and resource availability to decide whether the application traffic flow is admitted or rejected. For the case with no explicit control path involve-ment, the control message/protocol (for example, RTSP, H.323, SIP, HTTP, etc.) is not terminated or processed on the router 12. In this case, the system uses NBAR2, flow metadata information, etc., to extract information on the applicationflows. The system may, for example, sniff the SIP (Session Initiation Protocol), H.323, RTSP (Real-Time Streaming Pro-tocol), HTTP (Hypertext Transfer Protocol) content, and the like, to extract information for the application flows. If there is insufficient bandwidth resource for the flow, the traffic flow is marked as best effort or unadmitted class. Appropriate feedback messages are sent to the source of the application traffic flow to regulate the bandwidth consumption by these flows ( e.g., quench the traffic from source or lower the video bit rate to the minimum available bandwidth).")<br><br>Olofsson '254 at Abstract ("A method for routing is disclosed. The method comprises establishing an overlay network, comprising a plurality of network elements and an overlay controller; wherein the overlay controller is in communication with each network element via a secure tunnel established through an under-lying transport network; receiving by the overlay controller, information from each service-hosting network element information said information identifying a service hosted at that service-hosting network element, and label associated with the service-hosting network element; identifying by the overlay controller, at least one policy that associates traffic from a site with a service; and causing by said overly controller, the at least one policy to be executed so that traffic from the site identified in the policy is routed using the underlying transport network to the service-hosting network element associated with the said service.")<br><br>Olofsson '254 at Figure 1 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 1*<br>Olofsson '254 at Figure 3 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | |  <br>*FIG. 3*<br><br>Olofsson '254 at 2:27-44 ("In one embodiment, to realize the service chain construct, network elements may be interconnected across a regular network infrastructure in order to provide an overlay net-work on top of the regular network infrastructure. FIG. 1 shows an embodiment 100 of the overlay network. Referring to FIG. 1, the overlay network 100 includes an overlay controller 102, a mapping server 104, and a plurality of overlay edge routers 106. The overlay controller 102 is configured to orchestrate the overlay network 100 |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | using a secure transport (TLS, Transport Layer Security, IETF RFC5246) and a designated overlay control plane protocol over underlying network infrastructure 108. In one embodiment, the network infrastructure 108 may include a public network such as the Internet. The overlay control plane protocol may operate in a similar fashion to BGP (IETF RFC4271), in functions related to route and policy distri-bution, reliable transport over TCP (IETF RFC793), and optimal path selection process and distributed state creation.")<br><br>Olofsson '254 at 3:3-14 ("In one embodiment, within the overlay network 100, the overlay controller 102 processes control plane traffic, but does not get involved in the processing of data traffic. All data traffic is processed by the network elements present at site locations, such as a branch office, or central locations, such as a data center or a headquarters location. These network elements if, at a branch location is referred to as an "edge" and if, at a central location, is referred to as a "hub". In FIG. 1 hubs are indicated by reference numeral 110, whereas edges are indicated by reference numeral 106. In one embodiment, secure peer-to-peer links between the hubs and services define a forwarding plane, as shown in FIG. 2.")<br><br>Olofsson '254 at 3:64-4:9 ("In one embodiment, the overlay controller may be pro-visioned with or at least have access to traffic policy functions. These traffic policy functions may be distributed to selected hubs and edges and may be used to direct traffic. In one embodiment, the use of labels that identify services and provide for a forwarding tag, allows the overlay network 100 to overcome all of the previously presented challenges. Labels that represent Virtual Private Networks (VPN) may be combined, in some embodiments, with the Service labels to provide services that are VPN-specific and are reached using VPN-specific policies, versus general overlay network policies for reaching a service identified solely by a service label applicable to the entire overlay network.")<br><br>Olofsson '254 at 4:34-42 ("Based on the advertisements of routes from each edge and hub router and the advertisements of service labels from each hub router hosting a service, potentially restricted on a per-VPN basis by associating a service-label with a VPN-label, the overlay controller 102 constructs policies that are subsequently distributed to the network elements (hubs and edges) involved. The set of policies and their required contents are |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | related to the exact nature of the service chain that is being constructed. Two examples are provided below.")<br><br>Olofsson '254 at 6:36-59 ("Establishment of a Service Chain In one embodiment, to a method for establishing a service chain is shown in FIG. 3. Referring to FIG. 3, the method includes the following blocks:<br><br>Block 300: The overlay controller 102 establishes secure control channel with all associated network elements (hubs and edges).<br><br>Block 302: The service-hosting locations (hubs) advertise their service type and associated label to the overlay controller 102.<br><br>Block 304: The overlay controller 102 uses the service information received when constructing policies for the edge routers that are to use them.<br><br>The central controller can either: Apply the service policy to overlay routes before sending those to edge nodes with overlay next hop and label changed to that of service.<br><br>Block 306: The central controller pushes the service policies to the edge routers. These policies link traffic to the ultimate destination with a service chain.<br><br>Block 308: The central controller can also push policies to the service hosting routers, instructing them of their role in a given service chain and how to forward inbound and outbound traffic related to each VPN and each Service.")<br><br>Olofsson '254 at 7:22-28 ("In one embodiment, each edge node uses existing destination routes that are given a next-hop TLOC pointing to the entry point of a service chain. This route to TLOC assignment can be done by the central controller as a way of enforcing central service-chain policy, or by edge router when enforcing policies either distributed by the central controller or created locally on the device.")<br><br>Kumar '739 at Abstract ("Presented herein are service-function chaining techniques. In one example, a service controller in a network comprising a plurality of service nodes receives one is configured to identify one or more service-functions hosted by each of the service |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | nodes. The service controller defines a service-function chain in terms of service-functions to be applied to traffic in the network and provides information descriptive of the service-function chain to a classifier node.)<br><br>Kumar '739 at 3:66-4:12 ("The service controller 20 comprises service-function chaining logic 70 and the classifier 30 comprises classifica- tion and mapping logic 75. The service nodes 35, 40, 45, 50, and 55 each comprise advertisement logic 80. In operation, the advertisement logic 80 at each of the service nodes 35, 40, 45, 50, and 55 is configured to generate an advertisement or notification that indicates the service-functions that the respective service node hosts (i.e., each service node exposes its service-functions to the central service control-ler). For example, the advertisement logic 80 at service node 35 may generate an advertisement 85 indicating that the service node 35 hosts service-functions f1, f2, and f3. The advertisement 85 may then be provided to service controller 20 and/or classifier 30")<br><br>Kumar '739 at 4:13-26 ("The service-function chaining logic 70 at service control-ler 20 is configured to define one or more "service-function chains" (SFCs) for selection by the classification and map- ping logic 75 of classifier 30. As used herein, a "service-function chain" is an ordered list of service-functions defined in terms of the service-functions to be applied, and not in terms of service nodes that apply service-functions (i.e., the service-function chain is not defined in terms of network addresses for devices that host service-functions). More specifically, the location information of service nodes that host service-functions is not part of the defined service-function chain. Rather, as described further below, the selection of the location where the service-functions are available is performed at the classifier 30.")<br><br>Kumar '739 at 4:64-5:3 ("In the example of FIG. 2, classifier 30 intercepts traffic 90 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 90 should be steered through service-function chain SFC1 that comprises ordered service-functions f1, f2, f6, f7, and f10.") |

| No. | '111 Patent Claim 31 | The Reference |
|---|---|---|
| | | Kumar '739 at 6:21-31 ("As noted, service controller 20 may receive advertise-ments from service nodes 35, 40, 45, 50, and 55. Based on these advertisements, the service controller 20 determines that service-function f7 is hosted at service node 40. The service controller 20 also determines that service-function f7 is a modifying service-function (i.e., service-function f7 is capable of changing the flow specification of processed traffic). In the example of FIG. 3, service controller 20 defines service-function chains that include service-function f7 to account for the capability of service-function f7 to change the flow specification.")<br><br>Kumar '739 at 6:56-62 ("In the example of FIG. 3, classifier 30 intercepts traffic 150 for steering through a service-function chain defined by service controller 20. Using information (e.g., rules, poli-cies, etc.) provided by service controller 20, classification and mapping logic 75 determines that traffic 150 should be steered through service-function chain SFC3 that comprises first sub-chain SFC3a and the second sub-chain SFC3b.")<br><br>Kumar '739 at 8:7-22 ("FIG. 5 is an example block diagram of service controller 20. It should be understood that a virtual controller would be a software-emulated or virtualized version of what is shown in FIG. 5, such as software running on commodity hardware in a data center. The service controller 20 includes one or more processors 510, memory 522, a bus 530 and a network interface unit 540. The processor 510 may be a micropro- cessor or microcontroller. The network interface unit 540 facilitates network communications between the service controller 20 and network nodes (e.g., classifiers, service nodes, etc.). The processor 510 executes instructions asso-ciated with software stored in memory 522. Specifically, the memory 522 stores service-function chaining software 550 that, when executed by the processor 510, causes the pro-cessor 510 to perform the service-function chaining opera-tions described herein.") |