

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.
Petitioner

v.

PORTSMOUTH NETWORK CORP.
Patent Owner

IPR2024-00505
U.S. Patent No. 6,917,986

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

Mail Stop Patent Board
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. GROUNDS FOR STANDING.....	1
III. OVERVIEW OF THE '968 PATENT	1
A. Technology Background	1
1. Spanning Tree Protocol (STP).....	1
2. Root Switch.....	4
3. Access, Core, And Edge Switches.....	5
4. Redundant Links	7
B. Purported Invention.....	9
C. Prosecution History	11
IV. LEVEL OF ORDINARY SKILL IN THE ART.....	11
V. CLAIM CONSTRUCTION	11
VI. RELIEF REQUESTED	12
VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE.....	12
A. Challenged Claims And Statutory Ground For Challenge.....	12
B. Gai-EX1005.....	12
C. Ground 1A: Claims 1-2, 4-7, 9-14, 16-19, And 21-24 Are Obvious Over Gai (Downstream Embodiment).....	25
1. Independent Claims 1 and 12.....	25
2. Independent Claim 13	47

3.	Independent Claim 24	52
4.	Dependent Claims 2 and 14	54
5.	Dependent Claim 4.....	55
6.	Dependent Claims 5 and 17	57
7.	Dependent Claims 6 and 18	59
8.	Dependent Claim 7.....	60
9.	Dependent Claims 9 and 21	62
10.	Dependent Claims 10 and 22	63
11.	Dependent Claims 11 and 23	65
12.	Dependent Claims 16 and 19	65
D.	Ground 1B: Claims 1-2, 4-7, 9-14, 16-19, And 21-24 Are Obvious Over Gai (Upstream Embodiment).....	68
1.	Independent Claims 1/12	70
2.	Independent Claim 13.....	87
3.	Independent Claim 24.....	90
4.	Dependent Claims 2/14.....	91
5.	Dependent Claim 4.....	91
6.	Dependent Claims 5/17.....	94
7.	Dependent Claims 6/18.....	94
8.	Dependent Claim 7.....	94
9.	Dependent Claims 9/21.....	95
10.	Dependent Claims 10/22.....	95
11.	Dependent Claims 11/23.....	95

12.	Dependent Claims 16/19.....	96
VIII.	DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....	96
A.	Discretionary Denial Under <i>Fintiv</i> Is Not Appropriate	96
1.	No evidence regarding a stay	97
2.	Parallel proceeding trial date	97
3.	Investment in the parallel proceeding.....	98
4.	Overlapping issues with the parallel proceeding	98
5.	Identity of parties	99
6.	Other circumstances.....	99
B.	Discretionary Denial Under 35 U.S.C. § 325(d) Is Not Appropriate Because The Examiner’s Material Errors Warrant Reaching The Merits Of This Petition	99
1.	<i>Advanced Bionics</i> Step Two: The Examiner Erred By Not Considering Gai And Relevant Teachings Therein	100
IX.	CONCLUSION.....	103
X.	MANDATORY NOTICES	104
A.	Real Party-in-Interest	104
B.	Related Matters.....	104
C.	Lead and Back-up Counsel and Service Information	104

TABLE OF AUTHORITIES

Pages(s)

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH,</i> IPR2019-01469, Paper 6 (PTAB Feb. 13, 2020).....	99, 101
<i>Apple Inc. v. Fintiv, Inc.,</i> IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020).....	96, 98
<i>Becton, Dickinson & Co. v. B. Braun Melsungen AG,</i> IPR2017-01586, Paper 8 (PTAB Dec. 15, 2017)	100, 101
<i>Dish Network L.L.C. v. Broadband iTV, Inc.,</i> IPR2020-01359, Paper 15 (PTAB Feb. 12, 2021).....	97
<i>In re Apple Inc.,</i> 979 F.3d 1332 (Fed. Cir. 2020)	97
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.,</i> 868 F.3d 1013 (Fed. Cir. 2017)	12
<i>PEAG LLC v. Varta Microbattery GmbH,</i> IPR2020-01214, Paper 8 (PTAB Jan. 6, 2021)	98
<i>Phillips v. AWH Corp.,</i> 415 F.3d 1303 (Fed. Cir. 2005)	11
<i>Sand Revolution II LLC v. Continental Intermodal Group – Trucking LLC,</i> IPR2019-01393, Paper 24 (PTAB June 16, 2020)	97

Statutes

35 U.S.C. § 314.....	96
35 U.S.C. § 314 (a)	1
35 U.S.C. § 325(D)	99
35 U.S.C. §102(b)	12
35 U.S.C. §103(a).	1

35 U.S.C. §311 1

Rules

37 C.F.R. § 42.100(b) 11

37 C.F.R. §42.100 1

37 C.F.R. §42.104(a).....1

PETITIONER’S EXHIBIT LIST

1001	U.S. Patent No. 6,917,986 to Mor et al. (“’986 patent”)
1002	Prosecution History of U.S. Patent No. 6,917,986
1003	Declaration of Dr. Henry Houh
1004	<i>Curriculum Vitae</i> of Dr. Houh
1005	U.S. Patent No. 6,032,194 to Gai et al. (“Gai”)
1006	U.S. Patent No. 7,848,264 to Gai et al. (“Gai_264”)
1007	“IEEE Standard for Information Technology, Telecommunications and Information Exchange between Systems, Local and Metropolitan Area Networks, Common Specifications, Part 3: Media Access Control (MAC) Bridges,” published as ANSI/IEEE Standard 802.1D (1998) (“802.1D”)
1008	Docket Control Order, <i>Portsmouth Network Corp. v. Cisco Sys., Inc.</i> , No. 2:23-cv-00441 (E.D. Tex. Dec. 18, 2023)
1009	Director’s June 21, 2022, Memorandum on Discretionary Denials
1010	Prosecution History of U.S. Patent No. 7,848,264
1011	U.S. Patent No. 6,330,229 to Jain et al. (“Jain”)
1012	U.S. Patent No. 7,200,110 to Burns et al. (“Burns”)
1013	U.S. Patent No. 5,995,608 to Detampel et al (“Detampel”)
1014	U.S. Patent Publication No. 2003/0021223 to Kashyap (“Kashyap”)
1015	Notice of Disclosures, <i>Portsmouth Network Corp. v. Cisco Sys., Inc.</i> , No. 2:23-cv-00441 (E.D. Tex. Jan. 8, 2024)

Citations to specific pages of exhibits follow the pagination added to those exhibits per 37 C.F.R. § 42.63(d)(2)(i).

All emphasis has been added unless noted otherwise.

All annotations in Figures have been added unless noted otherwise.

I. INTRODUCTION

Pursuant to 35 U.S.C. §§311, 314(a), and 37 C.F.R. §42.100, Cisco Systems, Inc. (“Petitioner”) respectfully requests cancellation of claims 1-2, 4-7, 9-14, 16-19, and 21-24 (“Challenged Claims”) of U.S. Patent No. 6,917,986 (“’986 patent,” EX1001) as unpatentable under 35 U.S.C. §103(a).

II. GROUNDS FOR STANDING

Petitioner certifies that the ’986 patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR challenge. 37 C.F.R. §42.104(a).

III. OVERVIEW OF THE ’968 PATENT

A. Technology Background

1. Spanning Tree Protocol (STP)

“A computer network typically comprises a plurality of interconnected entities.” EX1005, 1:20-21. A network switch is a device that allows the network entities to exchange information, and “includes a plurality of ports that couple the switch to the other entities.” EX1005, 1:36-43.

Computer networks often provide redundant paths between the switches in the network to protect against failures such that a failure of any given link would not isolate any network portion. EX1005, 2:14-16 (“[C]omputer networks include redundant communications paths so that a failure of any given link does not isolate any portion of the network.”); EX1001, 2:7-9. The figure below illustrates two

along loops. The resulting traffic effectively overwhelms the network.”); EX1001, 2:23-29.

To avoid loops, networks execute the “spanning tree protocol” (STP) that blocks certain redundant paths to ensure only a single communication path exists. EX1005, 2:30-41 (“To avoid the formation of loops, devices, such as switches or bridges, execute a spanning tree algorithm. This algorithm effectively ‘severs’ the redundant links within the network. Specifically, ...the switches calculate [a] tree in accordance with the [STP] algorithm and typically elect to sever or block all of the redundant links, leaving a single communications path.”); EX1001, 2:30-47. For example, in the prior example, the STP algorithm can block certain ports (shown in “X”) leaving only a single communication path (in purple).

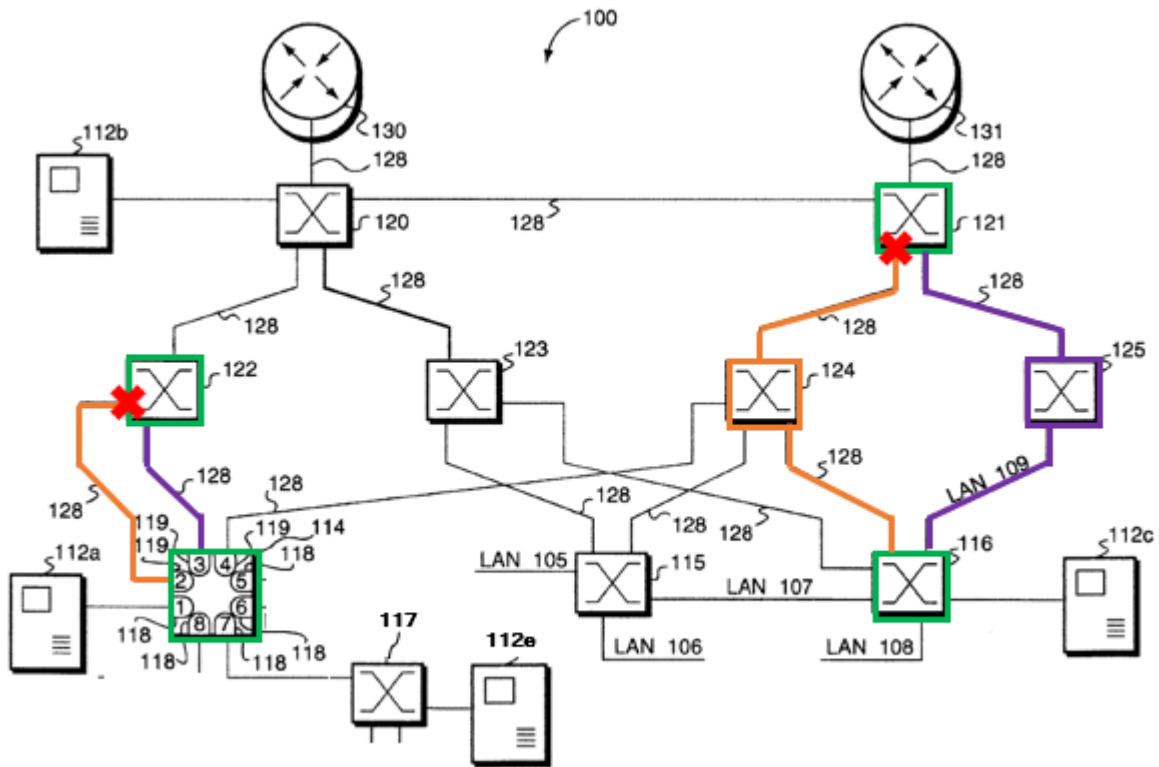


FIG. 1

EX1005, Figure 1; *see also*, EX1007, 76-103. EX1003, ¶¶34-37.

2. Root Switch

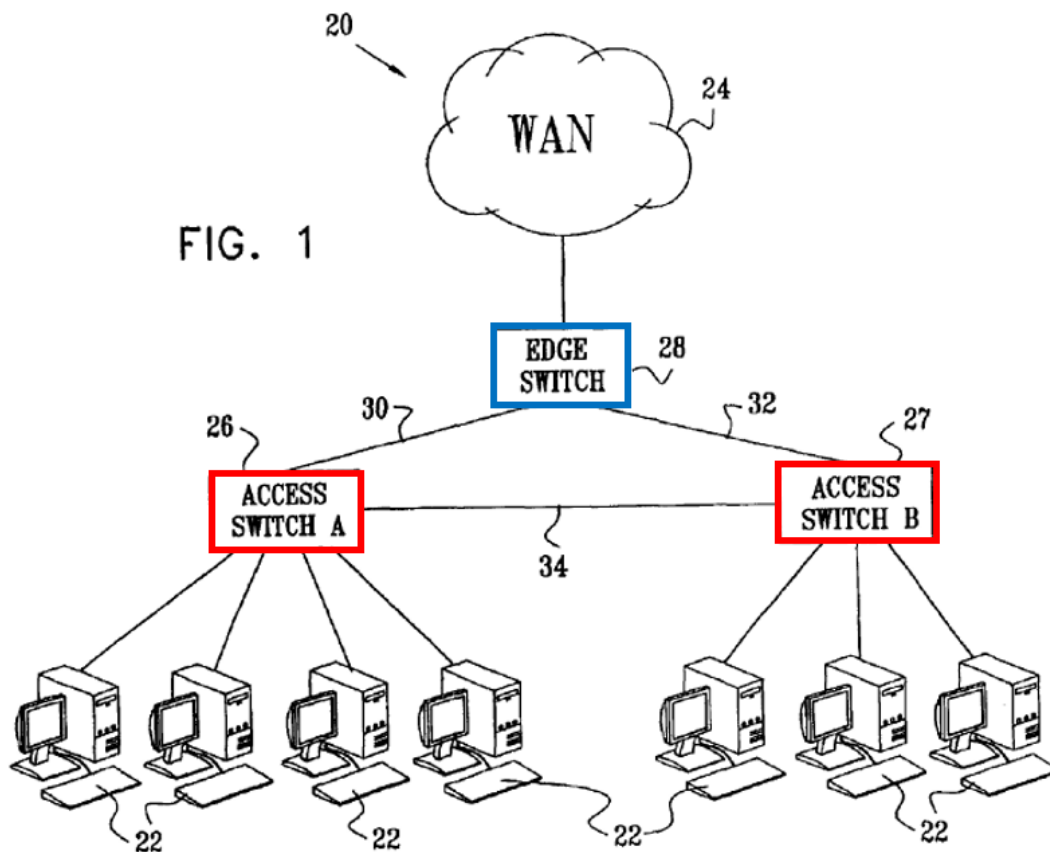
The STP algorithm can also elect one network switch as the “root” switch, which is the switch at the “root” of the tree. Any switch can be elected as the root switch. Switch 120 can be elected as the “root” switch, for example, if it is the switch with the lowest switch identifier or if it reduces the average distance between other switches (e.g., 121-125 and 114-116) and network entities (e.g., entities 112a, 112b, 112c, etc.) or based on a network administrator designation. EX1005, 2:42-47 (“[E]xecution of the spanning tree algorithm causes the switches to elect a single switch, among all the switches...to be the ‘root’ switch. Each switch has a unique

numerical identifier...and the root is the switch having the lowest switch ID numeric value..."); EX1005, 10:29-33 ("The conventional spanning tree algorithm, however, simply selects the switch having the lowest ID to become the root, regardless of [the] switch location. As a result, an access switch is often elected to be the root..."); EX1005, 3:17-39 ("Depending on the configuration of a given network, the location of the root can significantly affect the distance that messages must travel...To reduce the average distance [between end stations and the primary server] and avoid partitioning of the backbone switches, it is desirable to locate the root at a backbone switch..."); EX1001, 2:48-49; EX1005, 3:34-36 ("[A] network administrator may 'force' the network to choose the selected switch as the root."). EX1003, ¶38.

3. Access, Core, And Edge Switches

Switches can be "classified as access switches or core switches." EX1001, 1:30-31. "An access switch has ports that connect to LANs and/or to other network endpoints, such as user computer systems." EX1001, 1:31-33. "Core switches interconnect the access switches into a Wide Area Network (WAN), by direct connection between the core switch and access switches and/or by connection to other core switches." EX1001, 1:33-36. Moreover, "[c]ore switches that are directly connected to access switches are referred to herein as edge switches." EX1001, 1:36-38.

For example, **switches 26/27** are access switches because they have ports that connect to LANs and/or to other network endpoints (computer systems 22). **Switch 28** can act as a root, core, and an edge switch. *First*, it can be the root, if it has the lowest switch identifier. *Second*, it is a core switch because it interconnects **access switches 26/27** into a WAN by direct connection between the core switches and access switches and/or by connection to other core switches. *Third*, **switch 28** is also an *edge switch*, because it is a core switch that is directly connected to an access switch, e.g., **access switches 26/27**.



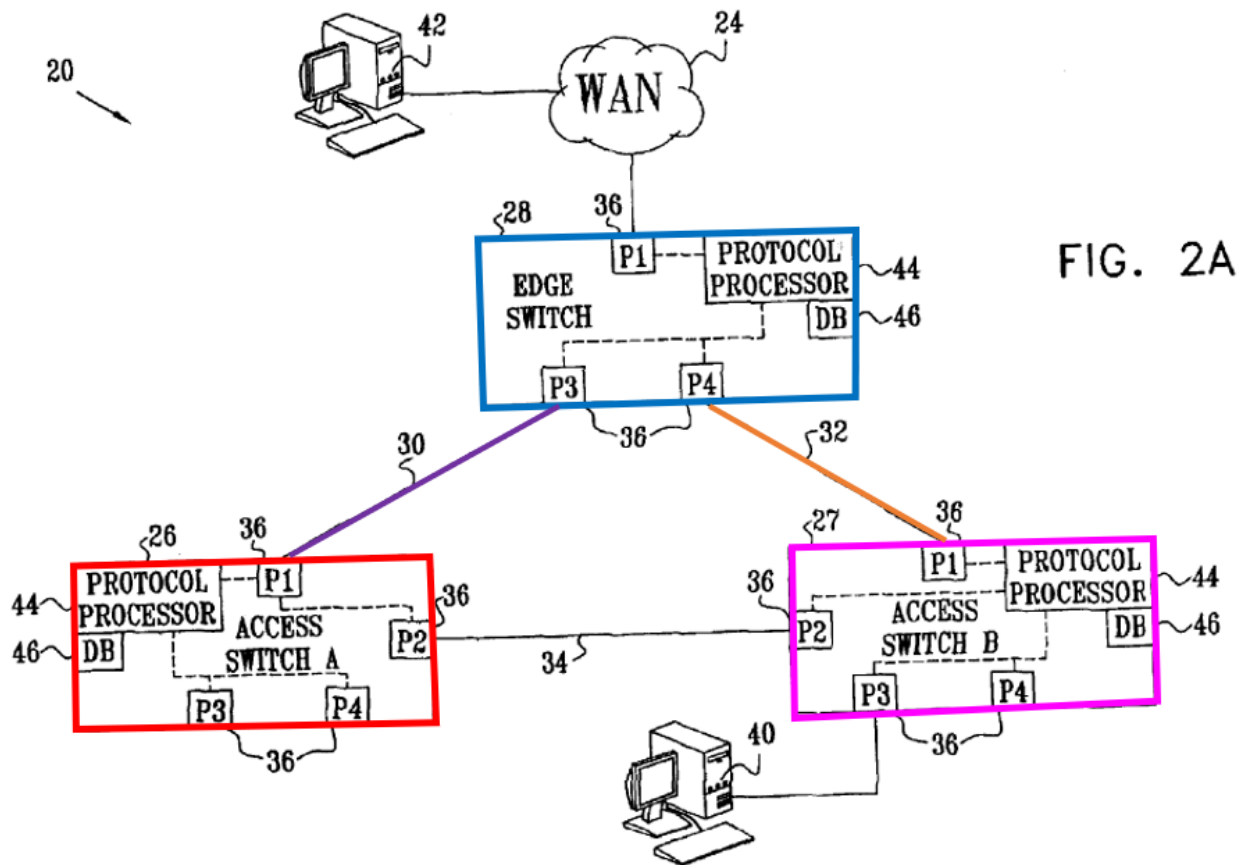
EX1001, Figure 1. EX1003, ¶¶39-40.

4. Redundant Links

“FIG. 1 is a block diagram [of] a computer network...which is configured with redundant links for fast protection of user access in case of network failure...” EX1001, 7:8-11. As shown, “[n]etwork 20 serves multiple users 22, who are connected via **access switches 26** and **27** (marked access switches **A** and **B**) to a WAN 24.” EX1001, 7:12-14. Further, “[e]ach of the access switches is connected by a respective **access link 30** or **32** to **edge switch 28**, which is in turn connected to the core WAN [24].” EX1001, 7:14-16. “A lateral link 34 between access switches [**26/27**] is used to provide protection in case of a failure in one of access links **30** or **32** or in the corresponding ports of the access switches or edge switch.” EX1001, 7:16-20.

“Each of the switches has multiple ports 36...as well as a protocol processor 44....” EX1001, 7:56-59. The ’986 Patent explains that “[P]rocessor 44...build[s] and update[es] a filtering database (DB) in a memory 46 and running STP.” EX1001, 8:1-3. In each switch, “the database includes entries associated with a plurality of the ports...” EX1001, 5:33-34.

Additionally, “processor 44 in **edge switch 28** is responsible for managing redundant links **30** and **32**,” (EX1001, 8:1-5), and can “select[] one of links **30** and **32** to serve as its active link to access switches **26** and **27**, while the other link is placed on standby.” EX1001, 8:11-13.



EX1001, Figure 2A.

For example, the '986 Patent assumes **link 30** as the active link. EX1001, 8:13-14 ("For the sake of example, let **link 30** be the active link.") The '986 Patent further describes that:

Processor 44 accordingly puts port P3 of **switch 28** in the forwarding state, while putting port P4 in blocking state...Therefore, when STP is run by processors 44, no traffic passes through **link 32**, and thus no loop exists, so that STP maintains port P1 of **switch 27** in the forwarding state. Since **switch 27** will receive no frames on port P1, it will also create no entries for this port in its filtering database.

EX1001, 8:14-24.

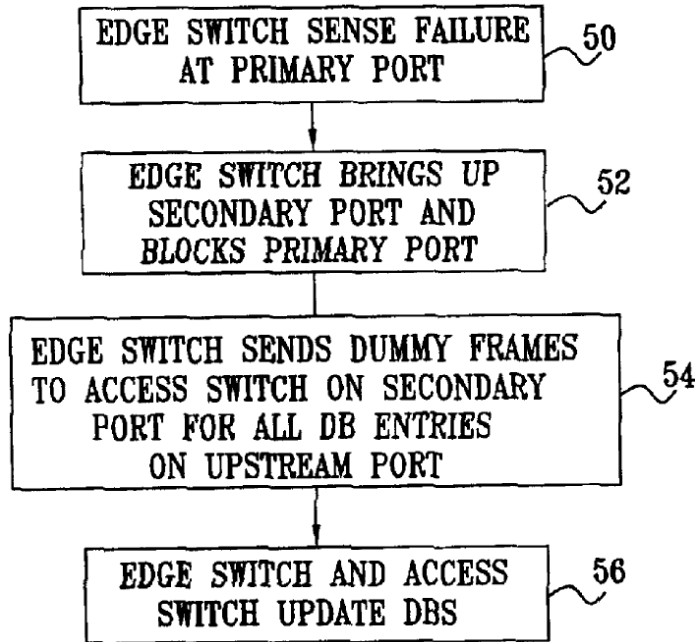
Because no traffic passes through **link 32**, all traffic directed to **switch 27** passes through **link 30** and link 34. EX1001, 8:62-64 (“[W]hen users 40 and 42 attempt to communicate over network 20, all the traffic between them will pass through **switch 26**, via links **30** and 34.”). EX1003, ¶¶41-45.

B. Purported Invention

The ’986 Patent purportedly provides “methods and devices for providing reliable, fault-resistant network access.” EX1001, 1:6-8.

Figure 3 “is a flow chart [of] a method by which **edge switch 28** responds to a failure in **link 30**...” EX1001, 9:11-13. The method is invoked when processor 44 senses a failure at the primary port P3 (step 50). *See* EX1001, 9:14-16. “Upon sensing the failure, processor 44 of **edge switch 28** blocks its [primary] port P3 and activates its secondary port P4 instead...” (step 52) EX1001, 9:25-30.

FIG. 3



EX1001, Figure 3.

Then, the “**edge switch 28** sends dummy frames downstream to **access switch 27** via **link 32**...(step 54). For every entry in the filtering database of the **edge switch** that is not associated with the failed port (P3 in the present example), the **edge switch** sends a dummy frame over **link 32** to **access switch 27**...”. EX1001, 9:42-48. “When **switch 27** receives the dummy frame, it also forwards it to **switch 26**.” EX1001, 9:56-57.

“Once the dummy frame has been sent, switches [**26/27/28**] update their filtering databases to reflect the change in the active link from **link 30** to **link 32**,...(step 56). **Edge switch 28** simply replaces the failed port (P3) in the entries in

its database with the new active port (P4) that has taken its place.” EX1001, 9:61-67. EX1003, ¶¶46-49.

C. Prosecution History

The '986 Patent was filed on January 7, 2002. EX1001, 1. While EX1005 is not listed on the cover, an Information Disclosure Statement filed on March 1, 2002, listed EX1005, (EX1002, 46) and the Examiner initialed it. EX1002, 30. However, EX1005 was not used by the Examiner in a rejection. Instead, the Examiner issued a first-action Notice of Allowance on March 10, 2005. EX1002, 23.

IV. LEVEL OF ORDINARY SKILL IN THE ART

A POSITA would have had a bachelor's degree in computer science, computer engineering, electrical engineering, or in an equivalent field, or approximately two years of experience working in the field of information technology and networking as of January 7, 2002. Lack of professional experience can be substituted by additional education, and vice versa. EX1003, ¶72.

V. CLAIM CONSTRUCTION

Claim terms in IPRs are construed according to their “ordinary and customary meaning” to those of skill in the art. 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*). Petitioner submits that, for the purposes of this proceeding and the grounds presented herein, no claim term requires express

construction. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

VI. RELIEF REQUESTED

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims.

VII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Challenged Claims And Statutory Ground For Challenge

Ground	Claims	Basis
#1	1-2, 4-7, 9-14, 16-19, and 21-24	(§103) Gai

B. Gai-EX1005

U.S. Patent No. 6,032,194 to Gai et al. (“Gai”), filed on December 24, 1997, and issued on February 29, 2000, is prior art under at least 35 U.S.C. §102(b).

Gai—a Cisco patent—was discussed in the Background Section of the ’986 Patent. *See* EX1001, 3:15-41. “[Gai]...describes a method for rapidly reconfiguring a computer network, based on certain modifications to the standard STP.” EX1001, 3:15-18. Specifically, Gai is directed to a “method that rapidly reconfigures a computer network having a plurality of devices executing the spanning tree algorithm” (EX1005, Abstract) and “more specifically, to a method and apparatus for rapidly reconfiguring a computer network *following a network change.*” EX1005, 1:14-17. EX1003, ¶52.

Gai Implements An STP

Gai's Figure 1 shows exemplary "network 100 [that] comprises a plurality of [LANs] 102-109 each of which is coupled to one or more access switches 114-116. **Access switches 114-116** are interconnected with each other through a plurality of **backbone switches 120-125.**" EX1005, 7:7-9. "Each **access switch 114-116** and **backbone switch 120-125**, moreover, includes a plurality of ports which, if coupled to a LAN 102-109, are referred to as local ports 118 and if coupled to links 128 are referred to as trunk ports 119." EX1001, 7:14-18.

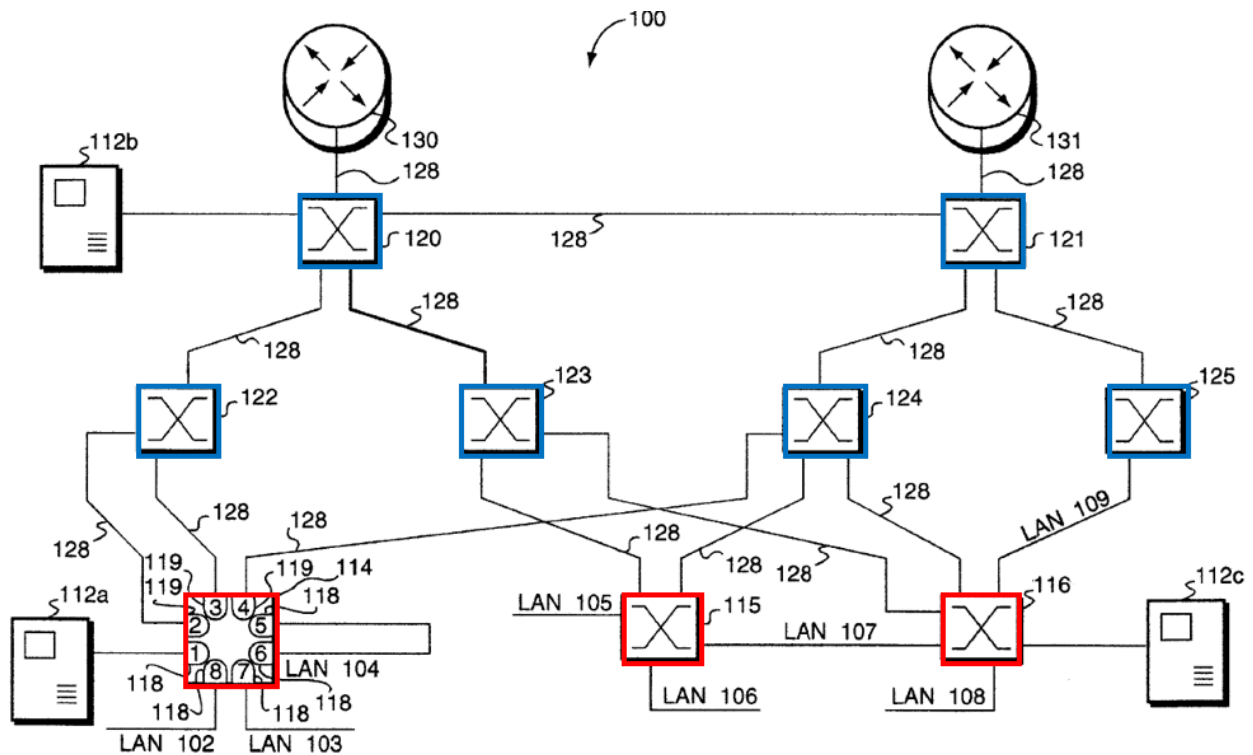


FIG. 1

EX1005, Figure 1. EX1003, ¶53.

In Gai, “**access switches 114-116** are generally connected to the **backbone switches 120-125** through a series of links 128, such as point-to-point links, although they may be connected via a shared media (e.g., LAN 109).” EX1005, 7:9-13. *See also id.*, 7:44-52 (“[O]ne or more distribution switches may be interleaved between the **backbone switches** and the **access switches**, two **access switches** may be directly connected, etc. ...[T]here is no distinction from the spanning tree point of view between local and trunk ports or point-to-point links and shared links (e.g., LANs). The distinction between local and trunk ports is made for purposes of explanation only.”). EX1003, ¶54.

Gai discloses connecting **access switches** to **backbone switches** using *redundant communication paths*. EX1005, 7:53-55 (“As shown, network 100 includes *redundant communication paths* interconnecting the access and backbone switches 114-116 and 120-125.”). The *redundant communication paths* provide alternative connectivity, but may result in loops, which are undesirable. EX1005, 7:55-59 (“The existence of such redundant links prevents portions of the network 100 from becoming isolated should any constituent link or device fail. Such redundancy, however, also results in the creation of loops, which...are highly undesirable.”). As such, Gai teaches “[t]o avoid the creation of loops, these switches execute the spanning tree algorithm...” EX1005, 7:59-61. EX1003, ¶55.

Similar to the '986 Patent, Gai explains that “[s]ignificantly, for each access switch 114-116, *only one port...that represents a path from the access switch to the root* (i.e., provides connectivity to the root through links, shared media, switches, etc.) *will be forwarding*. All other ports...that represent paths from the access switch to the root will be blocked. In other words, only one port at each access switch 114-116 that provides connectivity to the root will be forwarding.” EX1005, 11:8-15. EX1003, ¶56.

In the example below, for **access switch 114** having a path to root switch 121, the active port may be designated as port 3 (yellow circle). See EX1005, 11:41-44 (“With reference to switch 114, *for example*, assume that port number three, [is] in the forwarding state....”).

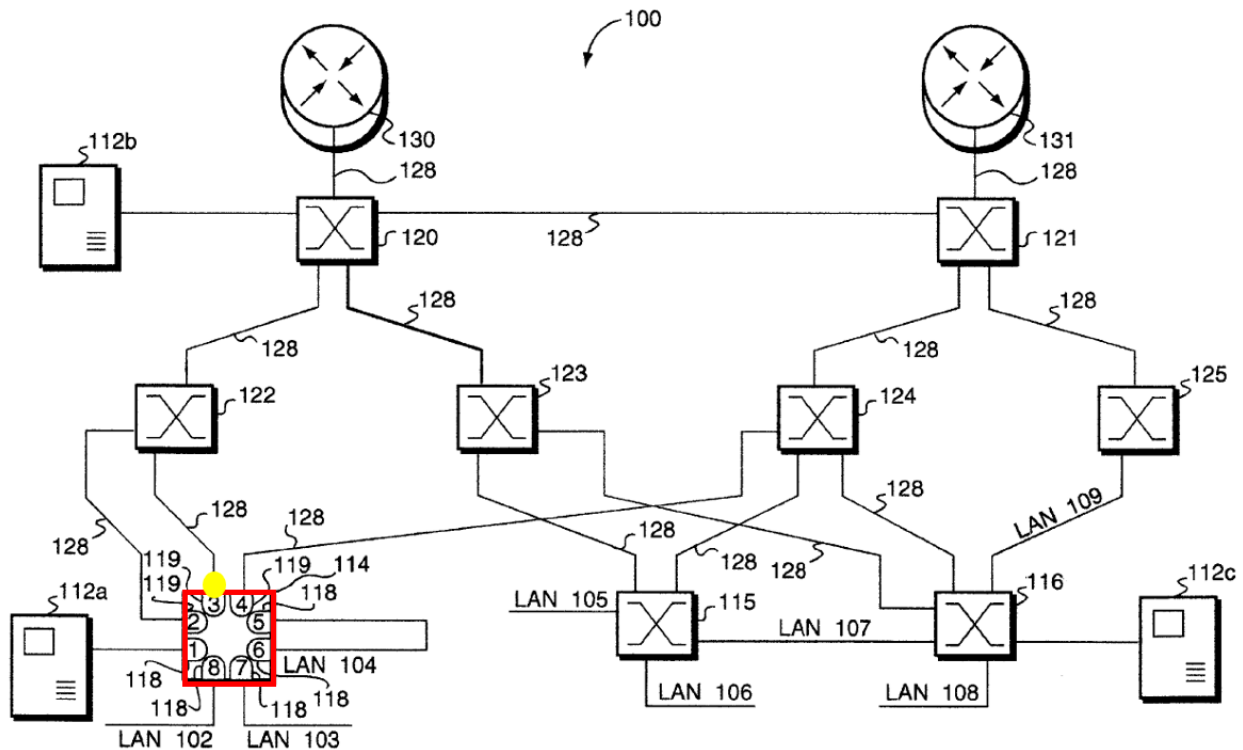


FIG. 1

EX1005, Figure 1. EX1003, ¶57.

Exemplary Failure In Gai's Access Switch 114

Gai explains that failures can happen to links or ports of switches. See EX1005, 2:14-16 (“[M]ost computer networks include redundant communications paths so that **a failure of any given link** does not isolate any portion of the network”); 4:1-3 (“If a switch stops receiving BPDUs on its root port, indicating a possible link or device failure, the corresponding timer will expire and the information will be discarded”); 5:36-47 (“The network preferably includes **a plurality of devices** executing the spanning tree algorithm so as to elect a root and place the ports of the

devices in either a forwarding or blocked state. In accordance with the method, *one or more devices* are configured and arranged so that one trunk port is in the forwarding state and other trunk ports are in the blocked state. Additionally, one or more of the blocked ports are designated as back-up ports. *Upon detection of a failure at the active forwarding port*, ...one of the back-up ports immediately transitions from blocked to forwarding, thereby becoming the new active port...”); 12:4-6. EX1003, ¶58.

Specifically, Gai provides one example of a failure at link 128 coupled to port 3 of access switch 114. EX1005, 11:36-38 (“Should a change occur in the network 100, *such as* a failure disabling the link coupled to a root port of an access switch...”); EX1005, 11:41-44 (“With reference to switch 114, *for example*, assume that port number three...is the root port for switch 114 and thus in the forwarding state.”); EX1005, 11:48-49 (“With reference to switch 114, *for example*, assume that port number three, [is] in the forwarding state....*If* the link 128 coupled to the port number three at switch 114 fails...”); EX1003, ¶59.

In this case, as the '986 Patent explains of Gai, “[u]pon detection of a failure at the active forwarding port, the switch immediately places one of the back-up trunk ports in the forwarding state, so that it becomes the new active port.” EX1001, 3:26-29. Specifically, either **port 2** or **4** (red dots) will become the active port of switch 114. EX1005, 11:48-52 (“If the link 128 coupled to the port number three at switch

114 fails, either one of these two other ports (i.e., port numbers two or four) will immediately transition to the forwarding state and begin receiving and sending messages.”). EX1003, ¶60.

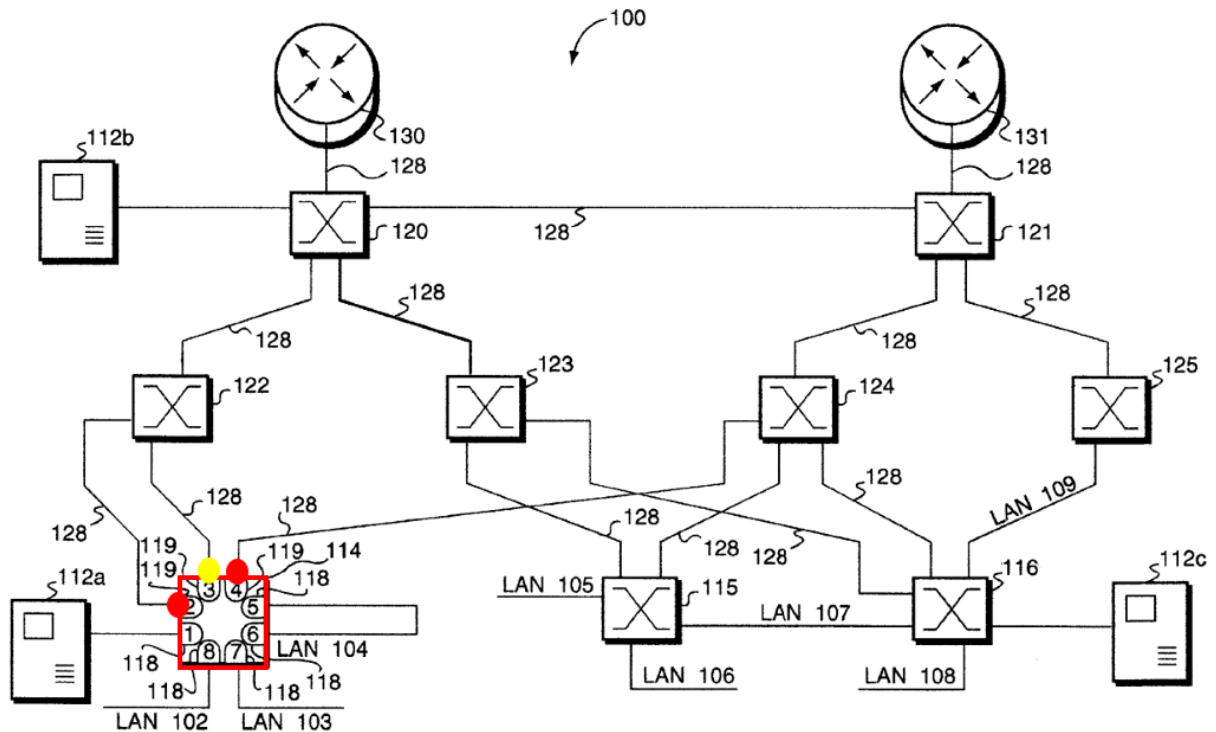


FIG. 1

EX1005, Figure 1.

Gai further discloses that “[u]pon transition to the new forwarding port, the device begins transmitting ‘dummy’ multicast messages through the new port” (EX1005, 5:54-56) to inform the network that a new topology exists for transmission of messages through access switch 114. Specifically, Gai explains that “[a]s ports transition between the blocked and forwarding states, entities may appear to move

from one port to another. To prevent switches from distributing messages based upon incorrect information, switches quickly age-out and discard the ‘old’ information in their filtering databases.” EX1005, 4:30-34. This is achieved by transmitting “dummy multicast message 400” and optionally “Topology Change Notification Protocol Data Unit (TCN-PDU) frames.” *See* EX1005, 15:12-18; EX1005, 4:34-37. EX1003, ¶61.

Specifically, Gai explains that “the transmission of dummy multicast messages 400, as described above, quickly informs devices of newly activated ports, thereby reducing the flooding that otherwise occurs when the contents of filtering databases 244 are discarded.” EX1005, 15:21-25. EX1003, ¶62.

Additional Embodiments of Gai Not Discussed In The ’986 Patent

To be clear, Applicant distinguished Gai from the ’986 Patent alleging that the ’986 Patent “manage[s] its redundant *downstream* ports in a novel way” (EX1001, 4:37-38), e.g., by “sending *dummy traffic...downstream*” (EX1001, 12:4-5), which is “in contrast to the methods described in [Gai], which require network entities to be aware of and manage redundancies in their *upstream ports*,” (EX1001, 4:39-42), e.g., by “transmit[ting] *dummy multicast messages* toward the root (*upstream*) through the newly-active trunk port.” EX1001, 3:30-31. EX1003, ¶63.

While this distinction is true when considering only one *example* of a failure on an *upstream* port and sending dummy traffic *upstream* (EX1005, 11:36-14:3), the claims of the '986 Patent are not so narrowly drafted. For example, only claims 11-12, and 23-24 recite such “***downstream***” transmission of dummy traffic. *See* EX1001, 11:53-55; 12:4-5; 13:7-14:15. Indeed, the scope of claim 1 includes sending dummy traffic upstream. EX1003, ¶64.

Indeed, as annotated below, Applicant did not discuss any of the other switches, such as **switch 124**, **access switch 115** and **access switch 116** that are similar to the '986 Patent's **edge switch 28**, **access switch 115** and **access switch 116**. *See, e.g.*, EX1001, 3:18-24 (“The method assumes that the switches in the network, including both *core and access switches*, have multiple trunk ports providing connectivity to the STP root via trunk switches. (These trunk switches are *comparable to the core switches and edge switches defined above...*).”).

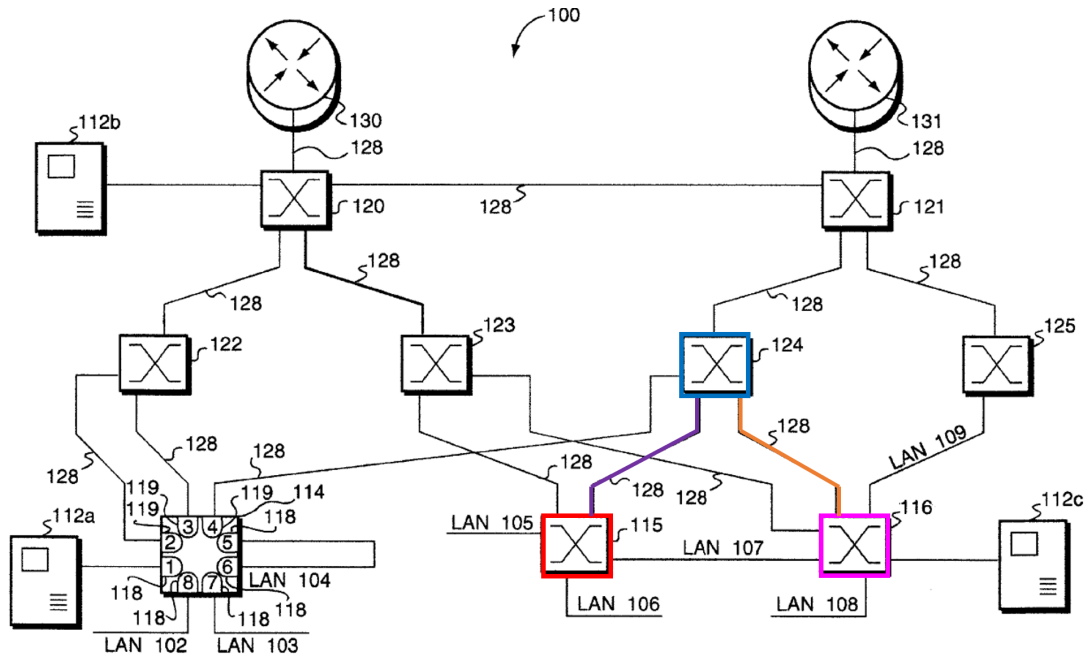
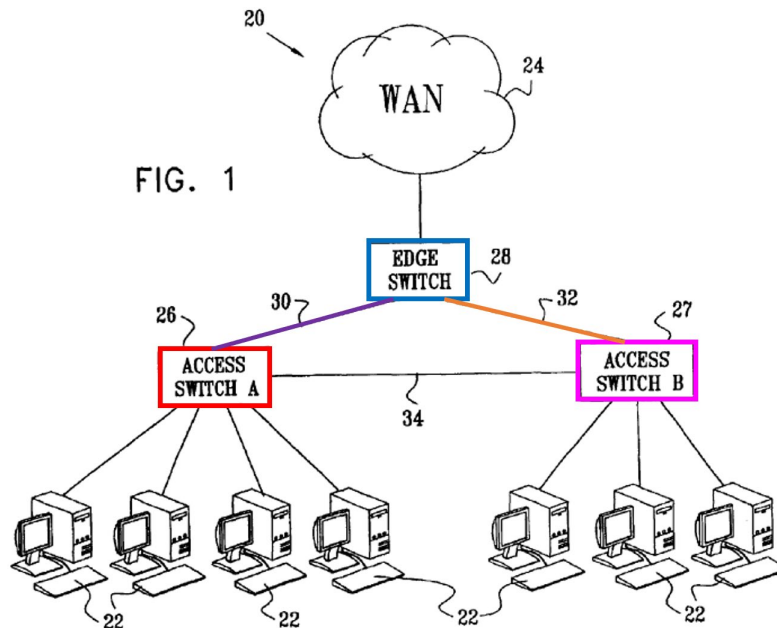


FIG. 1

EX1005, Figure 1. EX1003, ¶65.



EX1001, Figure 1. Just like the '968 Patent's **access switches 26** and **27**, switches **115** and **116** are **access switches** because they connect to LANs and/or to other

network endpoints, such as user computer systems. EX1001, 1:31-33. Just like the '968 Patent's **edge switch 28**, Gai's **switch 124** is an *edge switch* because it is a core switch¹ that is directly connected to **access switches 115/116**. See EX1001, 1:35-38. And similar to redundant links **30/32** of the '986 Patent, **first redundant link 128** couples **switch 124** to access **switch 115** and **second redundant link 128** couples **switch 124** to **access switch 116**. EX1003, ¶¶66-68.

Moreover, “assum[ing] that backbone switch 121...is the root” (EX1005, 14:16-17), the links identified with a **green arrow** indicate possible paths from **access switch 116** to the root switch (identified with a **purple arrow**). That is, one possible path includes sequentially **access switch 116**, **access switch 115**, **switch 124**, root switch 121, and the links (LAN 107 and links 128) coupling those switches. EX1003, ¶69.

¹ Because it interconnects access switches.

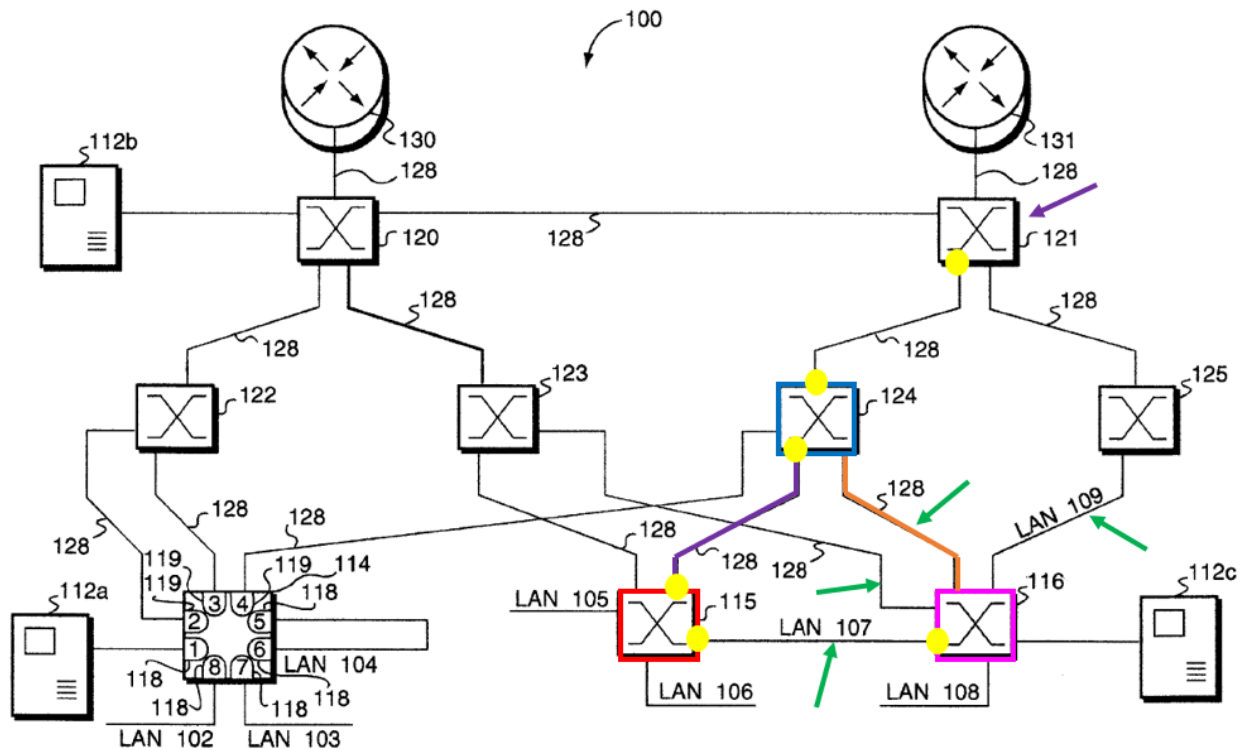


FIG. 1

EX1005, Figure 1. Accordingly, when the STP is implemented, only the port on the left side (yellow dot) of **access switch 116** will be forwarding, while the other ports of **access switch 116** will be blocked. See EX1005, 5:36-39 (“The network preferably includes a plurality of devices executing the spanning tree algorithm so as to elect a root and place the ports of the devices in either a forwarding or blocked state.”). And because Gai implements a spanning tree algorithm (i.e., there is only one path from the root to a network component), the ports identified with yellow dots and the corresponding links provide the path for a packet from the root port (yellow dot on switch 121) to the **access switch 116** and vice versa. EX1003, ¶70.

While Gai explicitly discusses the event of a failure in access switch 114, a POSITA would have readily recognized that failures can happen in any of the other switches, such as in one of the ports of **switch 124**. EX1012, 2:56-59 (“A...*port may fail for various reasons*, including a software defect, an equipment failure in the call control infrastructure or a failure in the transmission link.”). In that case, the active port that failed would be blocked (**purple X** below) and the port (**red circle** below) to **access switch 116** would be unblocked making **link 128** available for transmission. And a POSITA would have readily understood that Gai teaches (or at least renders obvious) sending dummy traffic through the newly-activated port (i.e., downstream) to inform the network that a new topology is now implemented.

and the remaining ports, providing connectivity to the root, are in the blocked state...” EX1005, Abstract. EX1003, ¶76.

Gai’s Figure 1 depicts an exemplary network:

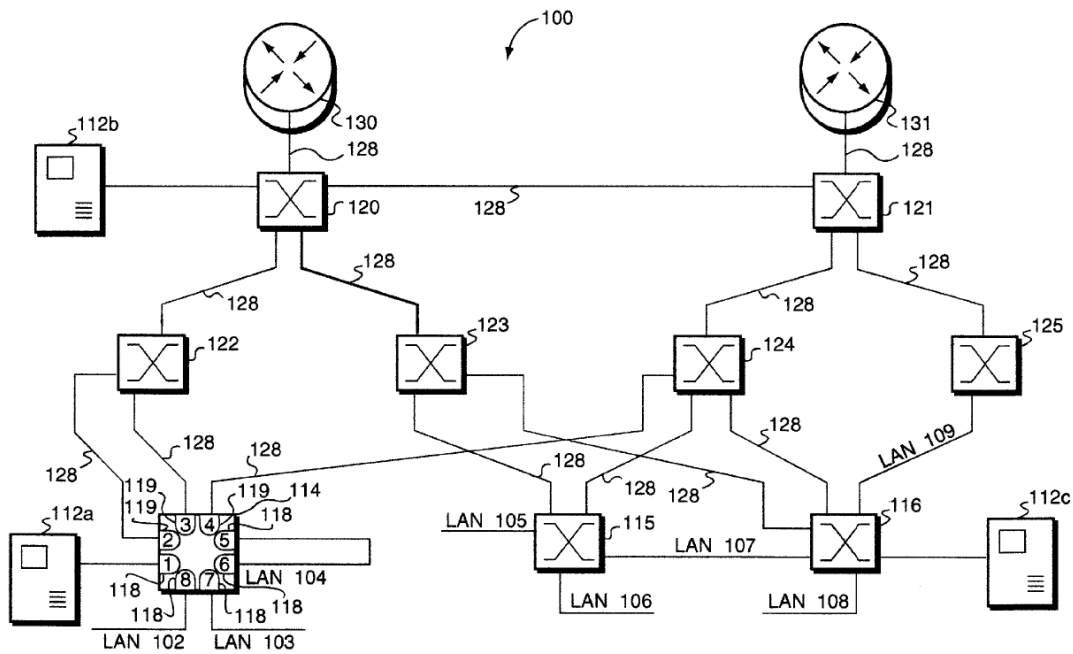


FIG. 1

EX1005, Figure 1.

- ii. [1a] coupling an **edge switch** in a network to one or more **access switches** via at least **first** and **second** redundant links, respectively connected to first and second ports of the **edge switch**;

[12a] coupling a **first bridge** in a network to a **second bridge** located downstream of the **first bridge**, by connecting at least **first** and **second** redundant links to respective first and second ports of the **first bridge** so as to communicate with the **second bridge**;

Gai discloses and/or renders obvious limitations [1a/12a]. At the outset, a “switch” is one particular type of a “bridge.” *See, e.g.*, EX1001, 1:26 (“The most common type of MAC bridge is a switch...”); 4:50-54. EX1003, ¶¶78-79.

Gai discloses an **edge switch** and one or more **access switches**, as the ’986 Patent admits. *See* EX1001, 3:15-24 (“[Gai] describes...switches in the network, including both core and **access switches**...[and] trunk switches. (These trunk switches are comparable to the core switches and **edge switches**....)”). Indeed, as shown below, Gai’s “**backbone**” **switch 124** meets the claimed **edge switch/first bridge** and Gai’s **access switch 115** and **access switch 116** meet the claimed **access switches/second bridge**. EX1003, ¶¶80-81.

Switch 124, shown below, meets the claimed **edge switch** because **edge switches** are “[c]ore switches that are directly connected to **access switches**.” EX1001, 1:36-37. *First*, **switch 124** is directly connected to **access switches**

115/116.² See EX1005, 7:9-14 (“**[A]ccess switches 114-116** are...connected to the **backbone switches 120-125** through...links 128, such as point-to-point links....”). Second, **backbone switch 124** is a core switch because it “interconnect[s] the **access switch [115/116]** into a [WAN], by direct connection between the core switch and access switches and/or by connection to other core switches [such as switch 121].” EX1001, 1:33-36; *see also* EX1005, 1:31-34 (“In many instances, several *LANs may be interconnected* by point-to-point links...*to form a wide area network (‘WAN’)*...”). Accordingly, Gai’s **backbone switch** corresponds to the claimed “**edge switch**”/“**first bridge**.” EX1003, ¶83.

² **Switches 115/116** have “ports that connect to LANs and/or to other network endpoints....” EX1001, 1:31-33; *see* EX1005, 7:4-7 (“The network 100 preferably comprises...[LANs] 102-109 each of which is coupled to one or more access switches 114-116.”); EX1005, 6:10-12 (“**Access switches** are preferably coupled to entities (e.g., LANs, end stations, etc.)...”). EX1003, ¶82.

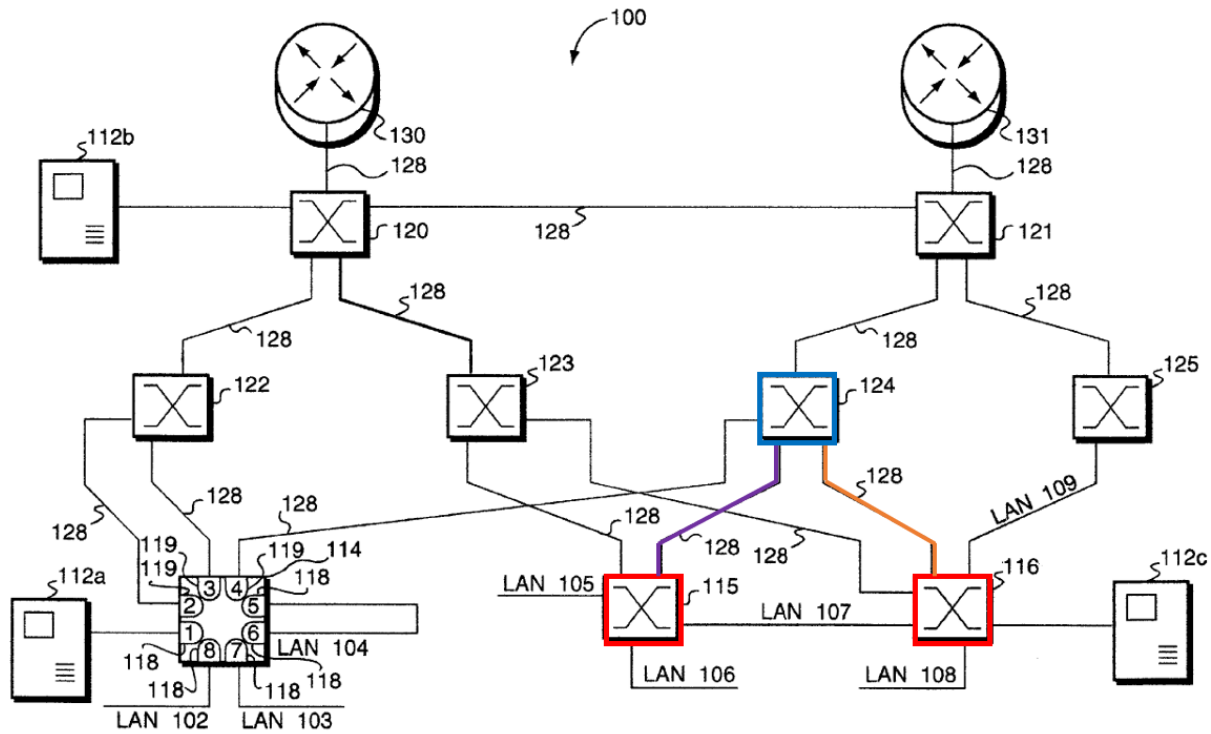
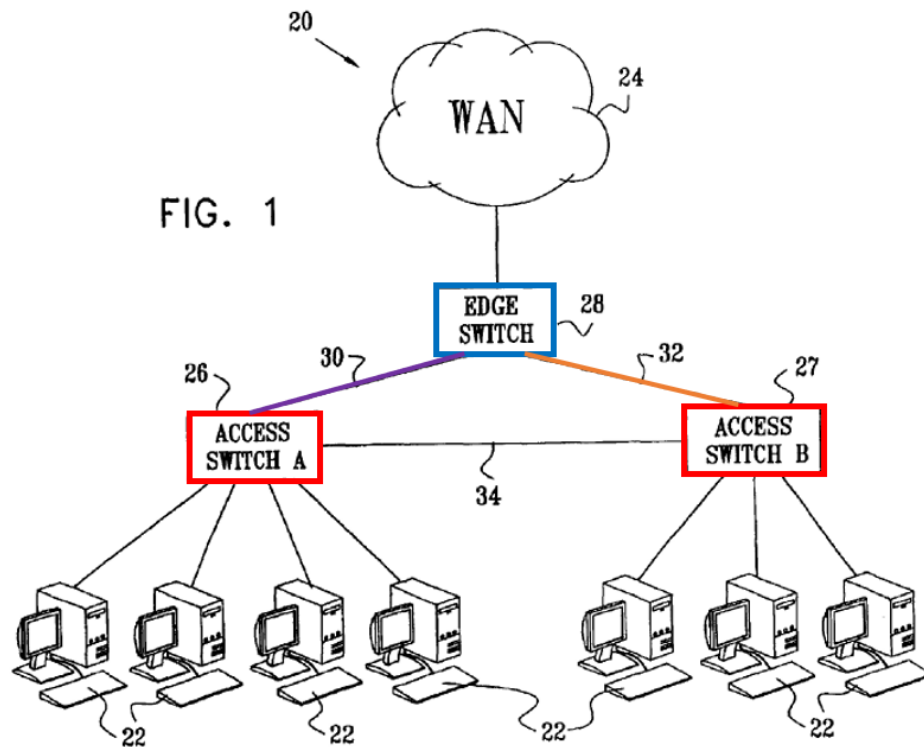


FIG. 1

EX1005, Figure 1.

Gai also discloses coupling the **edge switch** to the one or more **access switches** via at least **first** and **second** redundant links, respectively connected to first and second ports of the **edge switch**. As shown above, **switch 124** is connected to **access switches 115/116**, similar to Figure 1 of the '986 Patent showing **edge switch 28** connected to **access switches 26/27**. Moreover, in Gai, **first redundant link 128** couples **switch 124** to access **switch 115** so as to communicate with **switch 115** and **second redundant link 128** couples **switch 124** to **access switch 116**, just like redundant links **30** and **32** of the '986 Patent:



EX1001, Figure 1. EX1003, ¶84.

Although not explicitly shown in Figure 1, a POSITA would have understood that **switch 124** has ports, because ports are what are used to connect the switch with links, e.g., **first redundant link 128** and **second redundant link 128**. In fact, a POSITA would have understood that **switch 124** can have eight ports similar to switch 114 (circled **red**).

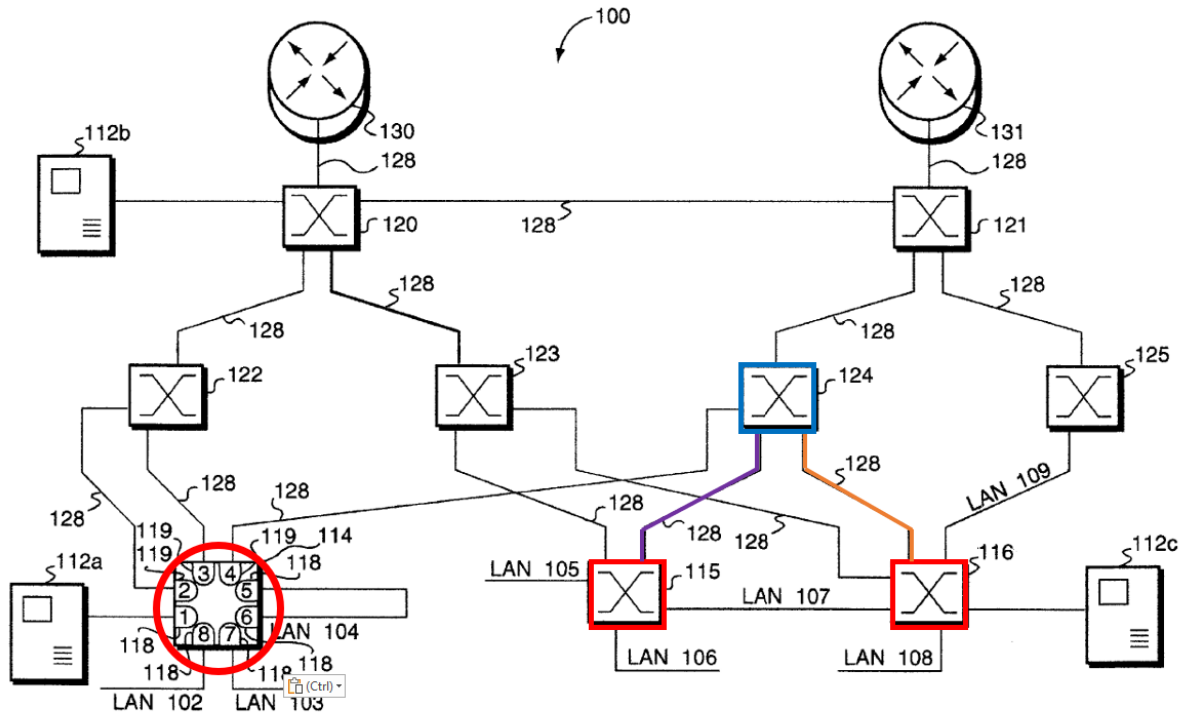


FIG. 1

EX1005, Figure 1. EX1003, ¶85.

Finally, **access switch [115]** [is] located downstream of **edge switch 124** (*i.e.*, away from root). See EX1001, 3:30-31 (“[T]oward the root (upstream)...”). For example, Gai assumes that **switch 121** is the root. EX1005, 14:16-17 (“For example, assume that backbone switch 121...is the root...”). Hence, **second bridge 115** is located downstream of **first bridge 124**. EX1003, ¶86.

- iii. [1b/12b] placing the first port in an **active state**, while placing the second port in a **blocking state**;

Gai discloses and/or renders obvious limitations [1b/12b]. The '986 Patent admits that Gai discloses this limitation. EX1001, 3:24-26 (“One of the trunk ports

of each switch is placed in the forwarding state, while the others are designated as back-up ports.”). In the example below, a **yellow dot** identifies the forwarding port and a **red dot** identifies the blocked back-up port.

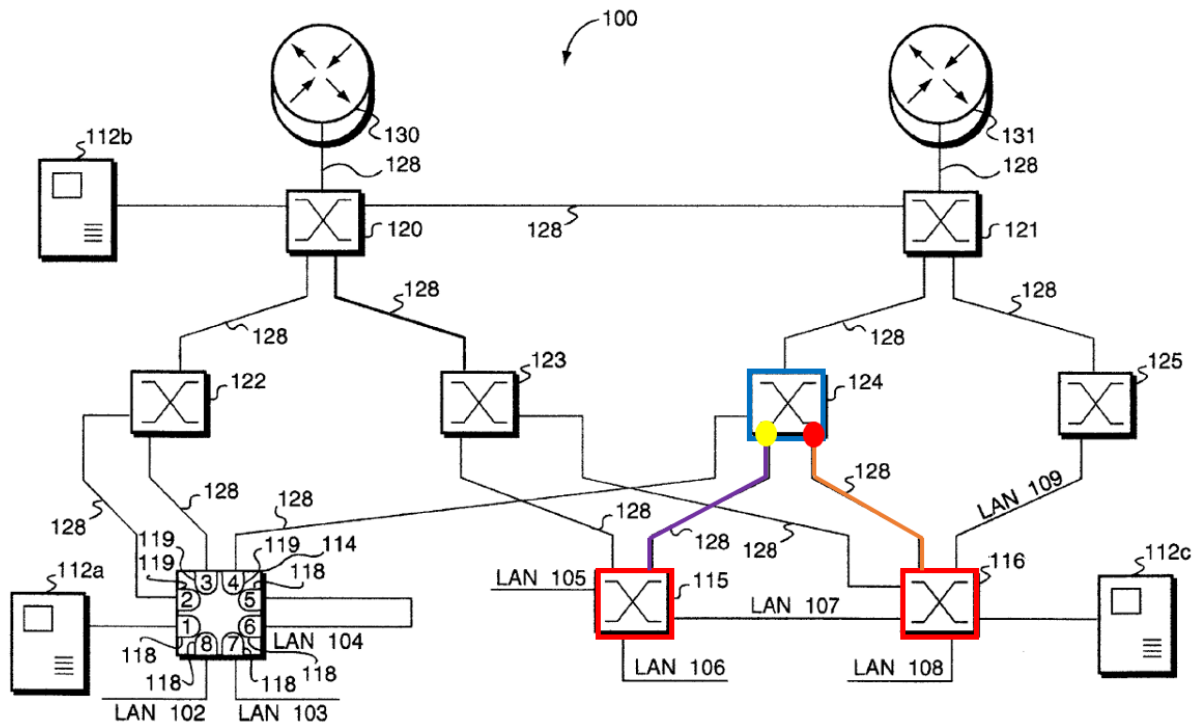


FIG. 1

EX1005, Figure 1. EX1003, ¶¶87-88.

Specifically, Gai discloses that “[t]he network...includes a plurality of devices executing the spanning tree algorithm so as to elect a root and place the ports of the devices *in either a forwarding or blocked state*...[D]evices are configured and arranged so *that one trunk port is in the forwarding state and other trunk ports are in the blocked state*....[O]ne or more of the blocked ports are designated as back-up ports.” EX1005, 5:36-44; *see also*, EX1005, Abstract. EX1003, ¶89.

Gai's network implements the STP. Accordingly, only the ports that are associated to links of the tree will be forwarding, while the other ports would be blocked. EX1005, 2:56-62 ("The root ports and designated switch *ports* are selected for inclusion in the spanning tree and are placed in a *forwarding state* so that data frames may be forwarded to and from these ports and thus onto the corresponding paths or links. *Ports not included within the spanning tree are placed in a blocked state.*"). EX1003, ¶90.

Accordingly, Gai teaches (or at a minimum suggests) that because the **first redundant link** is part of the STP, the first port (**yellow dot**) would be placed in an active state while the second port (**red dot**) tied to the **second redundant link 128** would be placed in a blocking state. EX1003, ¶91.

- iv. **[1c/12c] conveying communication traffic over the first link while the first port is in the active state, thereby causing [each of the switches/the bridges] to build a respective database for use in forwarding the traffic;**

Gai discloses and/or renders obvious limitations [1c/12c]. EX1003, ¶92.

First, Gai discloses "conveying communication traffic over the first link while the first port is in the active state." For example, Gai discloses that the ports that are part of the spanning tree are in a forwarding state, e.g., active state, so that data frames, e.g., communication traffic, can be conveyed to and from these ports and over the corresponding links:

The root ports and designated *switch ports* are selected for inclusion in the spanning tree and *are placed in a forwarding state* so that data frames may be forwarded to and from these ports and thus *onto the corresponding paths or links*. Ports not included within the spanning tree are placed in a blocked state. When a port is in the blocked state, data frames will not be forwarded to or received from the port.

EX1005, 2:56-63; *see also*, EX1005, 11:21-24 (“Assuming an entry is found and a corresponding destination port...is identified, the message is switched out onto this port, *assuming this port is in the forwarding state.*”). EX1003, ¶93.

Second, Gai discloses “thereby causing each of the switches/the bridges to build a respective database for use in forwarding the traffic.” For example, in Gai, switches inspect the last message received from a particular entity, identify which port it was received from, and save port information in a database; the port information will be used to reach the particular entity. EX1005, 1:50-54 (“Switches typically learn which destination port to use in order to reach a particular entity by noting on which source port the last message originating from that entity was received. *This information is then stored by each switch in a block of memory referred to as a filtering database.*”). Figure 2 below shows Gai’s filtering database.

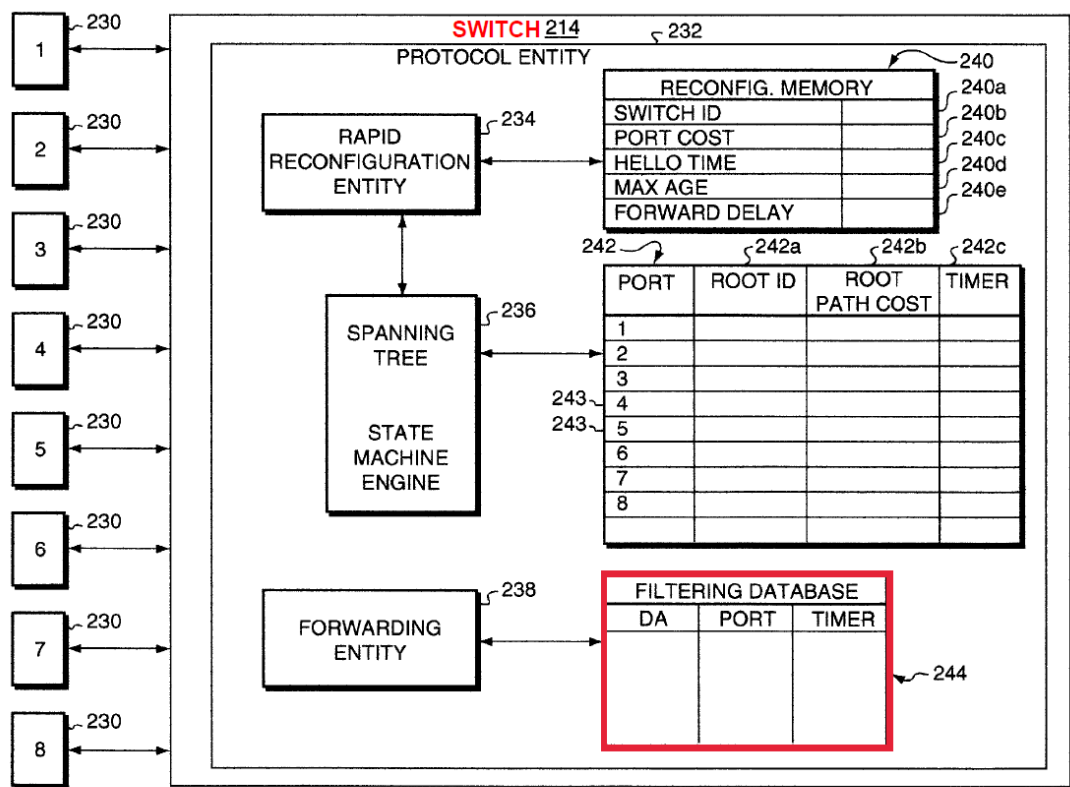


FIG. 2

EX1005, Figure 2. EX1003, ¶94.

Gai also discloses that the database port information is used when a message—received at the switch—is addressed to a particular entity; if no corresponding entry is already present, a new entry is formed. EX1005, 1:55-58 (“[W]hen a message addressed to a given entity is received on a source port, *the switch looks up the entity in its filtering database and identifies the appropriate destination port to utilize in order to reach that entity.*”); 11:21-24 (“Assuming an entry is found and a corresponding destination port...is identified, the message is switched out onto this port, assuming this port is in the forwarding state.”); 11:29-

33 (“If no entry is found for the source address, a new entry is formed and the source port...on which the message was received is entered as the destination port in the corresponding port column.”); 8:32-39; Fig. 2. EX1003, ¶95.

v. [1d/12d] responsive to a failure associated with the first link, placing the second port in the active state and the first port in the blocking state; and

Gai discloses and/or renders obvious limitations [1d/12d]. EX1003, ¶96.

The '986 Patent admits that Gai discloses this limitation. EX1001, 3:26-29 (“Upon detection of a failure at the active forwarding port, the switch immediately places one of the back-up trunk ports in the forwarding state, so that it becomes the new active port.”). EX1003, ¶97.

Indeed, Gai discloses that “[u]pon detection of a failure at the active forwarding port [indicated with **purple X**], the state of one of the back-up ports [**red dot**] immediately transitions from blocked to forwarding, thereby becoming the new active port for the device.” EX1005, 5:44-47; *see also*, EX1005, Abstract, 5:20-23, 11:36-41.

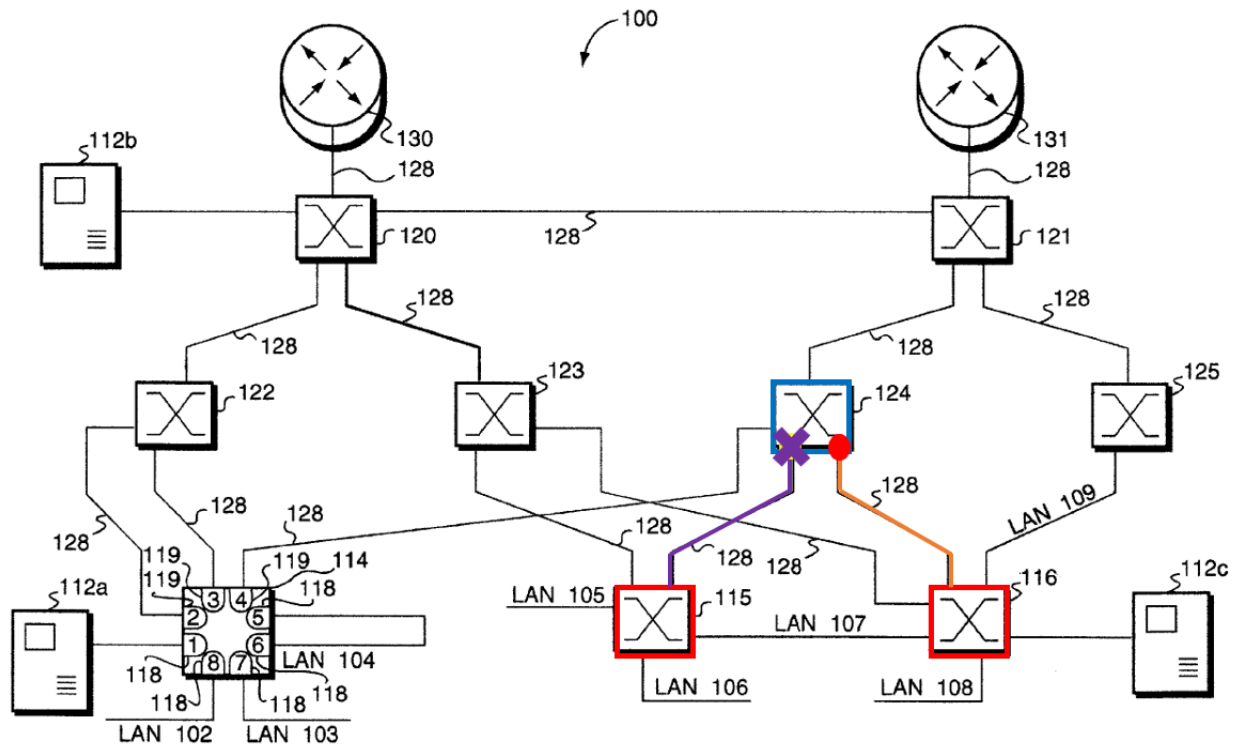


FIG. 1

EX1005, Figure 1. EX1003, ¶98.

A failure in a port is a failure associated with the link connected to the port, because the port is one endpoint of the link, and a failure on the port would result in a failure to the associated link. For example, Figure 3D shows “a flow diagram of a rapid reconfiguration process 340 following a *link failure*...In response to the detection of a *failure at port number three* (the root port), [block 342], rapid reconfiguration entity 234 at switch 214 *selects a backup port* to become the new root port, [box 344].” EX1005, 12:13-18; *see also* EX1005, 12:32-35 (“Rapid reconfiguration entity 234, at block 346, then directs the spanning tree state machine

engine 236 to immediately transition the selected back-up port...to the forwarding state.”). EX1003, ¶99.

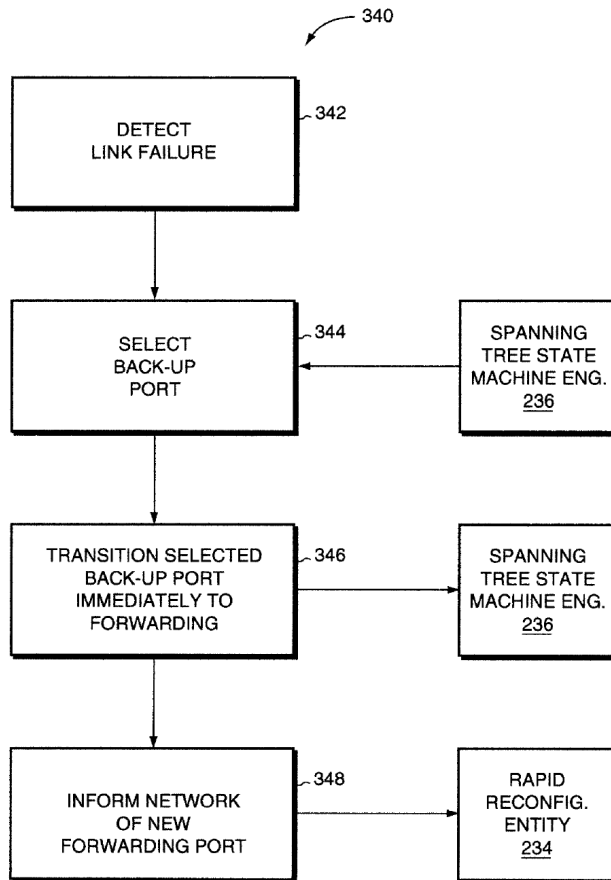


FIG. 3D

EX1005, Figure 3D.

- vi. [1e/12e] sending dummy traffic from the [edge switch/first bridge] over the second link [downstream] to [the one or more access switches/the second bridge], so as to cause each of [the one or more access switches/the second bridge] to modify its respective database responsive to the second port being in the active state.

Gai discloses and/or renders obvious limitations [1e/12e]. The '986 Patent admits that Gai discloses sending dummy traffic from a switch such that other switches in the network learn of the newly-active port. EX1001, 3:29-35 (“The switch [in Gai] then transmits dummy multicast messages toward the root (upstream) through the newly-active trunk port, each containing the source address of a network entity that is coupled to the switch itself or is downstream of the switch. Upon receiving these messages, other devices in the network learn to use the new active port.”). While the admission is explicitly directed to upstream transmission of dummy messages, as explained below, Gai also teaches (or at a minimum renders obvious) downstream transmission of dummy messages. EX1003, ¶100.

First, Gai discloses sending dummy traffic from the edge switch/first bridge over the second link downstream to the one or more access switches/second bridge. As the '986 Patent admits, Gai discloses one specific example of a failure associated with port 3 (marked with purple X) of switch 114 that transmits packets towards the root (upstream) through the newly activated port 2 (red dot). EX1005, 11:41-52 (“With reference to switch 114, *for example*, assume that port number three, which

is coupled to backbone switch 122, *is the root port* for switch 114 and thus in the forwarding state... *If the link 128 coupled to the port number three at switch 114 fails*, either one of these two other ports (i.e., port numbers two or four) will immediately transition to the forwarding state and begin receiving and sending messages.”), 12:62-66 (“Switch 114...informs the other devices of its new forwarding port, as indicated at block 348.”).

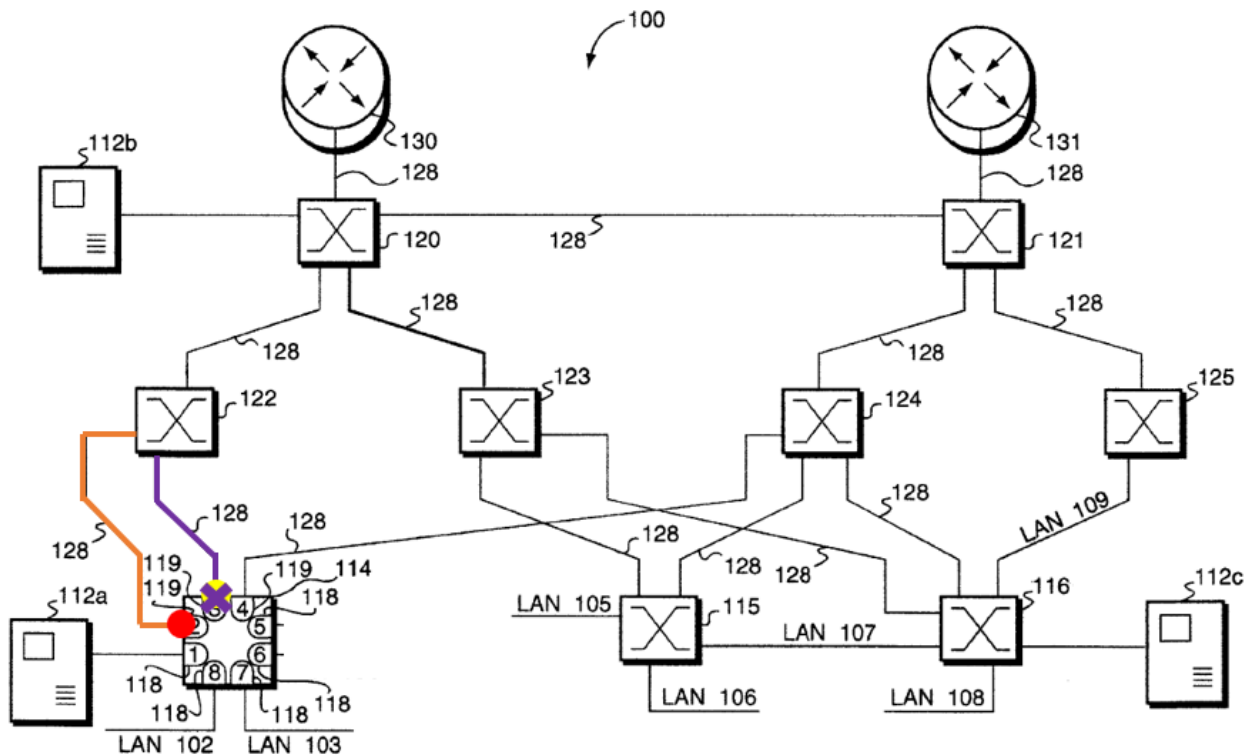


FIG. 1

EX1005, Figure 1. In this example, switch 114 is at the lowermost level and the particular port that failed happens to be a port transmitting packets directed towards

the root switch 121. EX1005, 14:16-17 (“For example, assume that backbone switch 121...is the root...”). EX1003, ¶¶101-102.

However, a POSITA would have understood that a failure can happen at any port or link. For example, failure can occur at a “designated” port, i.e., a port that transmits packets downstream. *See e.g.*, EX1012, 2:56-59 (“A...**port may fail for various reasons**, including a software defect, an equipment failure...or a failure in the transmission link.”); EX1013, 16:59-60 (“[The] bridge status memory file is updated **whenever any ports in any of said plurality of bridge servers fails**.”); EX1006, 17:61-63 (“[A] root port for communication with a root router and [a] **designated port to communicate with at least one router in a lower leaf of a spanning tree**.”); EX1011, 5:6-9 (“[I]mprovements to the spanning tree algorithm can be used to facilitate **rapid detection of a designated port** or designated bridge **failure**.”); EX1014, ¶63 (“**If any of the ports fails**, the...sub-switch...reconfigures the mappings to reroute the packets correctly.”). EX1003, ¶103.

For example, as shown below, if a failure happened at the port of **switch 124** that is connected to **first redundant link 128**, then a back-up port (indicated with a red dot) would transition from “blocked” to “forwarding” state, therefore coupling **switch 124** to **access switch 116** through **second redundant link 128**. *See e.g.*, EX1005, 5:44-47 (“Upon detection of a failure at the active forwarding port, the

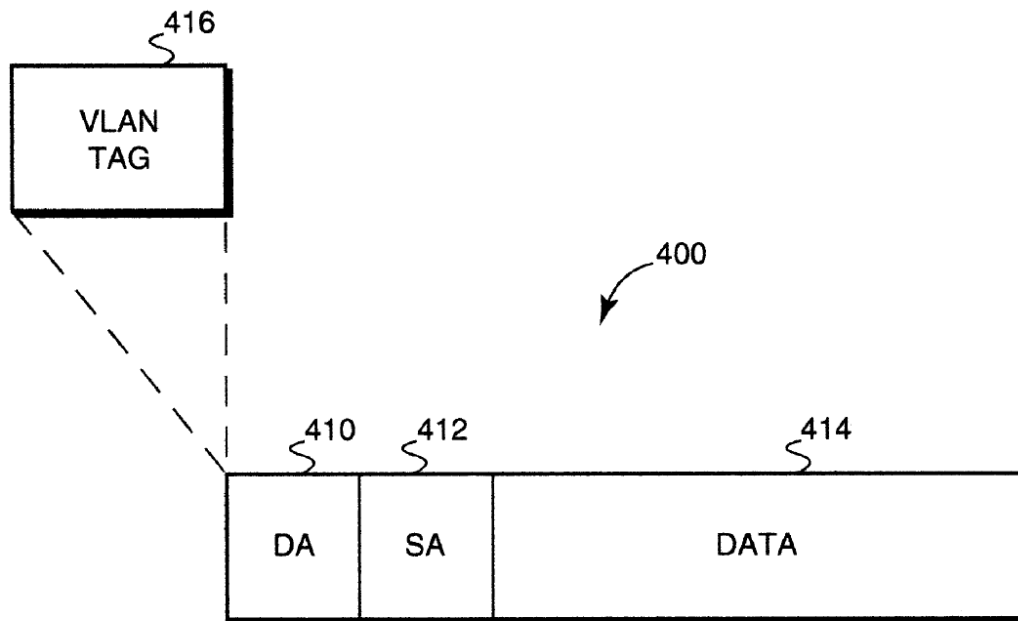


FIG. 4

EX1005, Figure 4; *see also*, EX1005, 12:66-13:1 (“FIG. 4 is a highly schematic illustration of a *dummy multicast message* 400...”). EX1003, ¶106.

Indeed, in U.S. Patent No. 7,848,264 (EX1006), a continuation application of EX1005, EX1006 claimed transmitting dummy messages downstream from a newly-active designated port, i.e., a port transmitting packets downstream, to inform devices downstream about the new designated port. EX1006, 18:8-11 (“[T]ransmitting, by the back-up designated port, one or more messages, *the messages to tell devices downstream in the spanning tree that the router has a new designated port.*”); *see also*, EX1006, 17:57-18:11. In fact, it was the Examiner that added the limitation “transmitting, by the back-up designated port, one or more

messages, the messages to tell devices *downstream* in the spanning tree that the router has a new designated port,” to claim 1 through an Examiner’s amendment to allow the claims. EX1010, 364-372. EX1003, ¶107.

And sending dummy traffic from a new active port downstream so “that other devices within the network [i.e., the downstream devices] learn to utilize the new forwarding port, rather than the failed port, when directing messages to these entities” (EX1005, 5:65-6:1), would have been consistent with the “object of the [Gai] invention to provide a method and apparatus for reducing the time necessary to reconfigure the network following a change...” EX1005, 5:16-19. EX1003, ¶108.

Second, Gai discloses and/or renders obvious “so as to cause each of the one or more *access switches/second bridge* to modify its respective database responsive to the second port being in the active state.” For example, Gai discloses that “[u]pon receipt [of the dummy messages,] the other devices examine the contents of these messages and note the port on which they were received, which may differ from the port on which messages from these entities were previously received (i.e., before the failure and subsequent replacement of the device's active port). It is through this process that other devices within the network learn to utilize the new forwarding port, rather than the failed port, when directing messages to these entities.” EX1005, 5:60-6:1; *see also* EX1005, 12:56-62, 13:66-14:3. EX1003, ¶109.

Gai discloses that the switch that receives the dummy message from the new port enters this information in its filtering database such that this new port information is used when the switch sends a message to the newly activated port. For example, in connection with the example of a failure at port 3 of switch 114, Gai discloses:

Since the multicast message 400 having the source address of server 112a was received by backbone switch 124 at *a new source port, this new location information is entered by backbone switch 124 into its filtering database*, replacing the previous information that was stored therein. Thereafter, if backbone switch 124 receives a message intended for server 112a, *it will use this new destination port* and the message will be received at port number four of switch 114, which is now capable of receiving and forwarding messages...Backbone switch 124 distributes the multicast message 400 through all of its forwarding ports *so that other devices, as necessary, may learn of the new forwarding port....*

EX1005, 13:34-49; *see also*, 12:66-13:33; 13:66-14:3. EX1003, ¶110.

Accordingly, a POSITA would have understood that when *access switch 116* receives a dummy message from the new port it would have updated its filtering database with the new port information and would also have distributed the dummy message to *access switch 115*, such that switch 115 also learns of the new forwarding port, e.g., by updating its corresponding database.

2. Independent Claim 13

i. [13pre] A **network switching device**, comprising:

To the extent the preamble is limiting, Gai discloses and/or renders obvious the preamble. Gai's Figure 1 discloses examples of a **network switching device**, such as **switch 124**.

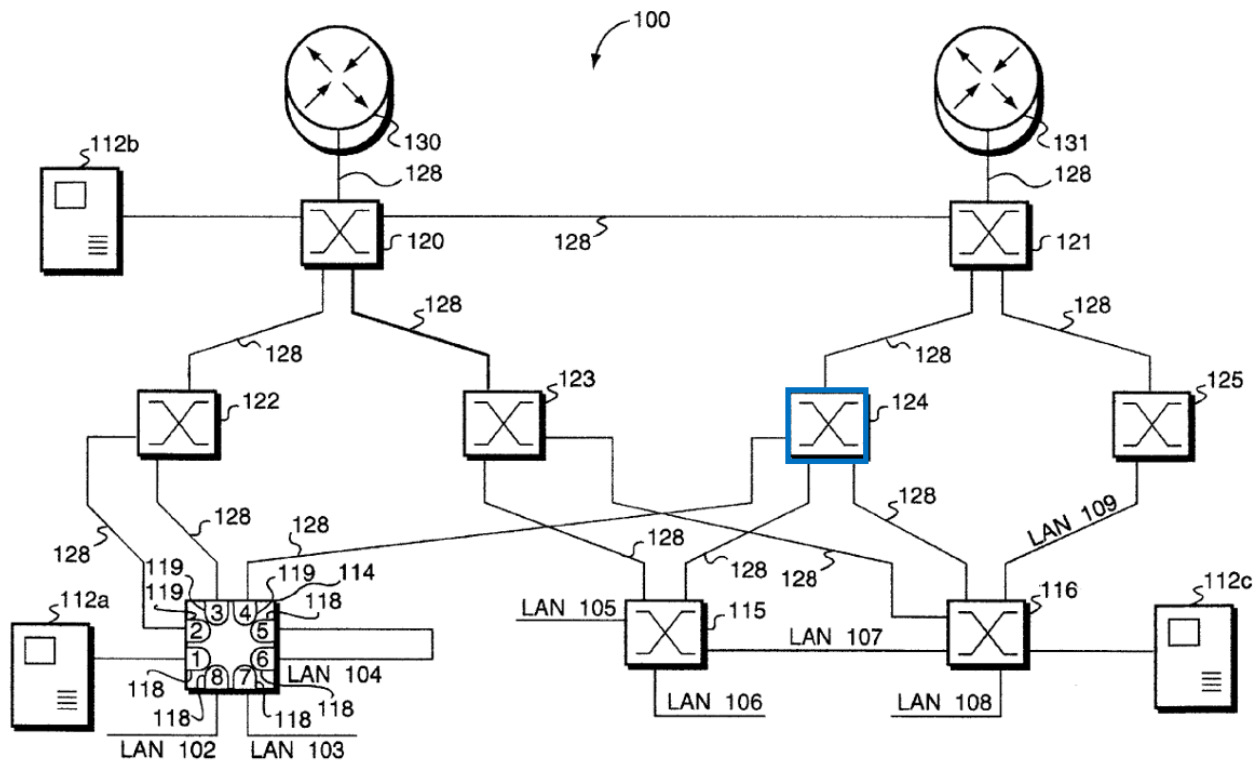


FIG. 1

EX1005, Figure 1. EX1003, ¶113.

- ii. [13a] a network port, adapted to be coupled over a network trunk link to a core switch in a network so as to exchange communication traffic with the core switch;

Gai discloses and/or renders obvious this limitation. For example, in the example below, *switch 124* has a network port (in yellow dot) coupled to core switch 121 over link 128 identified with a green arrow, and meets the claimed “trunk link.”

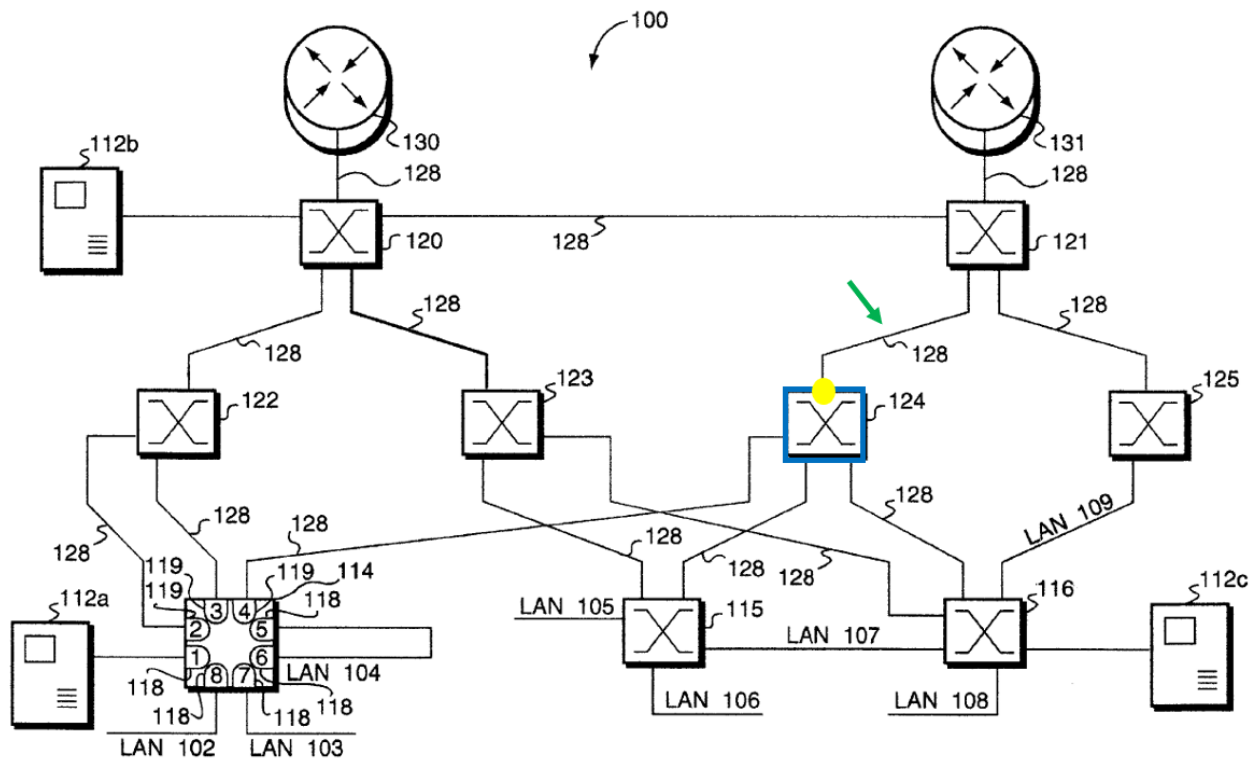


FIG. 1

EX1005, Figure 1. EX1003, ¶114.

Moreover, a POSITA would have also understood that the network port of *switch 124* exchanges communication traffic with core switch 121 because it is directly connected—through the link—to it. EX1003, ¶115.

- iii. **[13b] first and second access ports, adapted to be coupled via first and second redundant network access links to one or more access switches serving network users; and**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.1.ii. Gai’s “first” and “second” ports (limitations [1a/12a]) are “access” ports because they are connected to the downstream links to the access switches. EX1001, 4:1-2 (“[A]ccess ports (i.e., the ports that are connected to the downstream links to the access switches)...”). EX1003, ¶116.

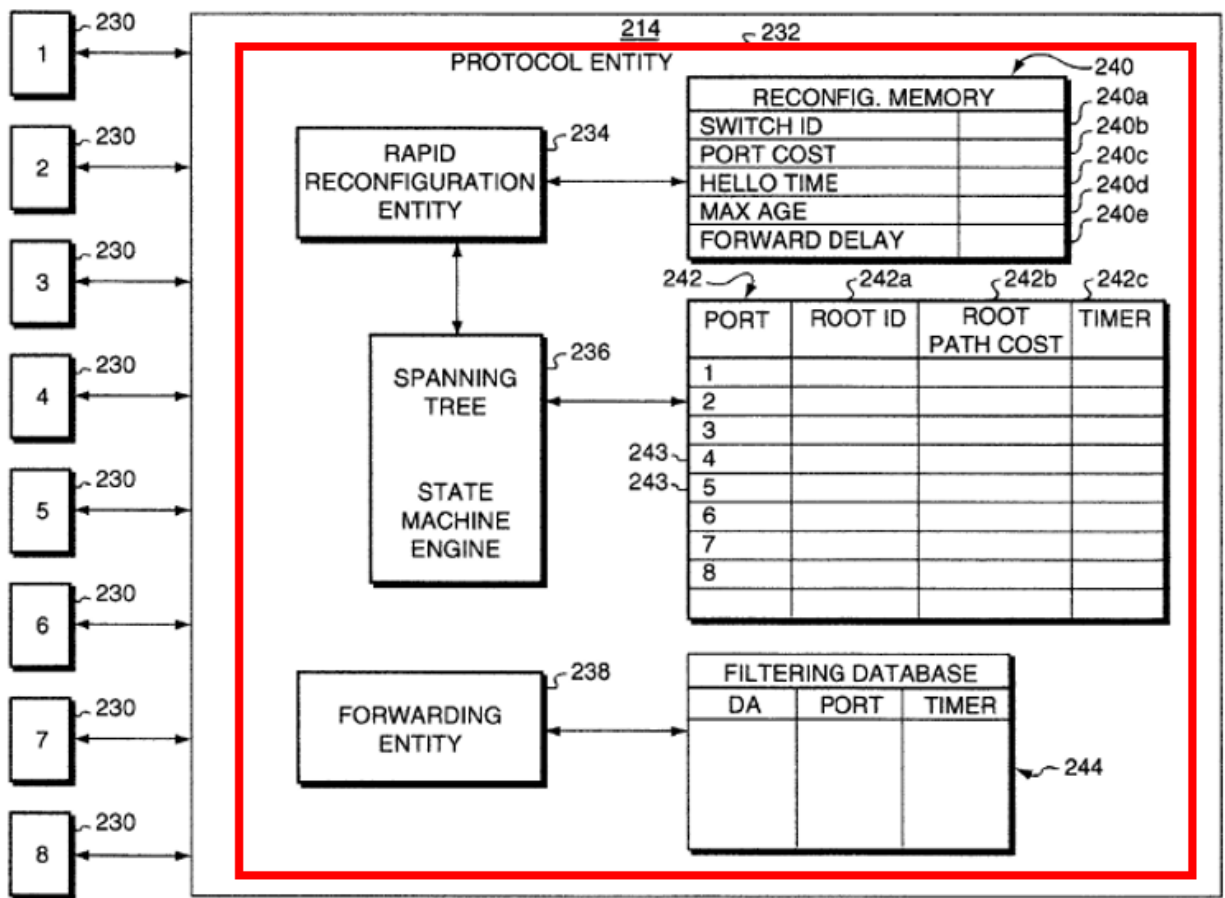
A POSITA would have also understood that Gai’s access switches serve network users. For example, a POSITA would have understood that users would be connected to the LANs shown in Figure 1. *See also*, EX1005, 5:2-7. EX1003, ¶117.

- iv. **[13c] a protocol processor, adapted to place the first access port in an active state, while placing the second access port in a blocking state,**

Gai discloses and/or renders obvious limitation [13c]. EX1003, ¶118.

First, Gai discloses a “protocol processor.” For example, Gai discloses “a plurality of devices executing the spanning tree algorithm” (EX1005, 5:37-38) where “switches exchange special messages called bridge protocol data unit (BPDU) frames that allow them to calculate a spanning tree or active topology...”. EX1005, 2:33-35. Specifically, Gai’s network switching device “includes *a protocol entity* 232 [that] includes a rapid reconfiguration entity 234, at least one spanning tree state

machine engine 236 and a forwarding entity 238.” EX1005, 8:12-15. A POSITA would have understood that protocol entity 232 is a protocol processor because it includes “processing elements” that implement the STP. EX1005, 8:46-50 (“[E]ntit[y] 234...preferably comprise[s] *programmable processing elements*”); 7:53-61 (“[A]ccess and backbone switches 114-116 and 120-125...execute the spanning tree algorithm.”).



EX1005, Figure 2. EX1003, ¶119.

Second, Gai discloses and/or renders obvious “adapted to place the first access port in an active state, while placing the second access port in a blocking state,” for the reasons discussed in Section VII.C.1.iii. EX1003, ¶120.

- v. **[13d] so that the communication traffic is conveyed over the first link while the first port is in the active state, thereby causing each of the access switches to build a respective access switch database for use in forwarding the traffic,**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.1.iv. EX1003, ¶121.

- vi. **[13e] the protocol processor being further adapted, responsive to a failure associated with the first link, to place the second access port in the active state and the first port in the blocking state and**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.1.v. In addition, Gai discloses that “[r]apid reconfiguration entity 234 [within protocol entity 232] then directs the spanning tree state machine engine 236 to immediately transition the selected back-up port...to the forwarding state.” EX1005, 12:32-35. EX1003, ¶122.

- vii. **[13f] [the protocol processor being further adapted, responsive to a failure associated with the first link] to send dummy traffic through the second access port over the second link to the one or more access switches, so as to cause each of the one or more access switches to modify its respective access switch database responsive to the second access port being in the active state.**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.1.vi. In addition, Gai discloses that “switch 114, via rapid reconfiguration entity 234 [within protocol entity 232], preferably informs the other devices of its new forwarding port by transmitting dummy multicast packets through the new port...” EX1005, 12:62-65. EX1003, ¶123.

3. Independent Claim 24

- i. **[24pre] A bridge device for use in a communication network having a spanning tree root, the device comprising:**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.C.2.i, VII.C.1.i, VII.C.1.ii, VII.C.1.vi. EX1003, ¶124.

- ii. **[24a] a designated port, adapted to be coupled over an upstream link in the network so as to communicate with the root;**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.2.ii. Additionally, *switch 124* has a designated port (yellow dot) that connects it via an upstream link to root bridge 121.

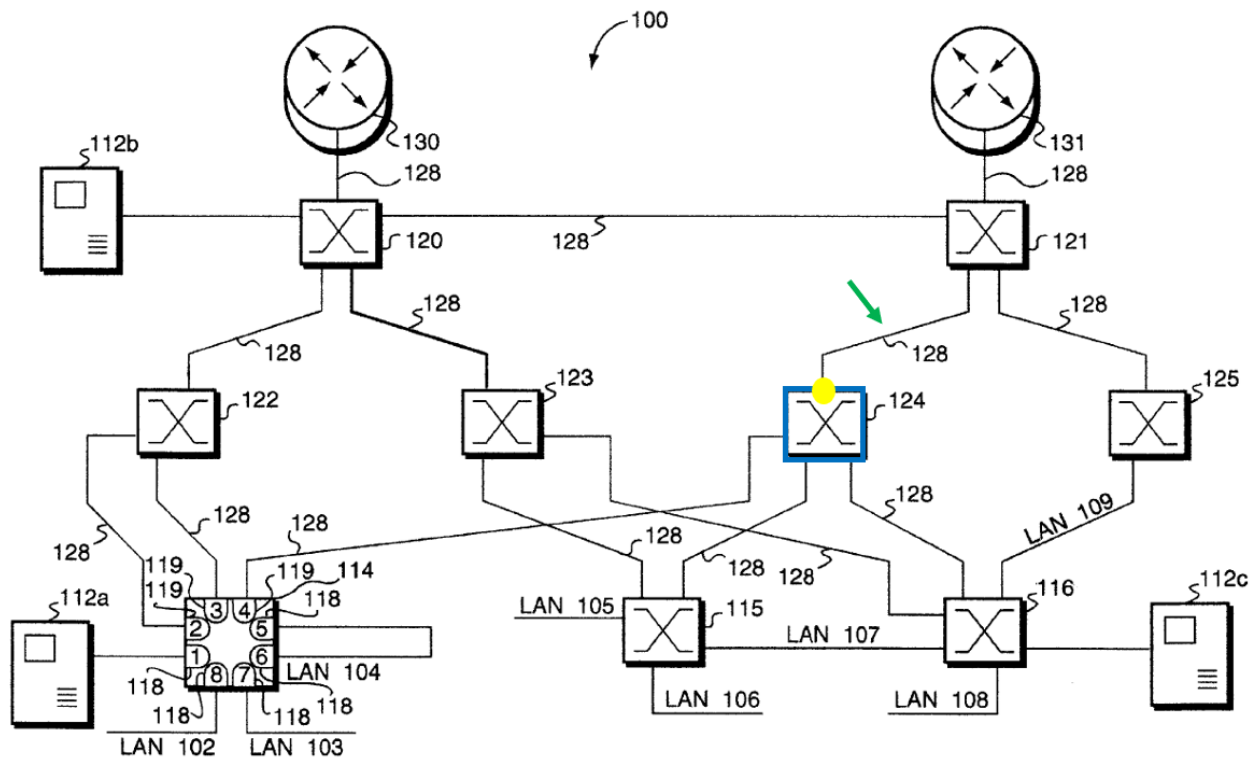


FIG. 1

EX1005, Figure 1. EX1003, ¶125.

- iii. [24b] first and second downstream ports, adapted to be coupled via first and second redundant links to a further bridge device located downstream in the network relative to the spanning tree root; and

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.C.1.ii and VII.C.1.vi. EX1003, ¶126.

- iv. **[24c] a protocol processor, which is adapted to place the first downstream port in an active state, while placing the second downstream port in a blocking state,**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.C.2.iv and VII.C.1.vi. EX1003, ¶127.

- v. **[24d] so that communication traffic is conveyed over the first downstream link while the first port is in the active state, thereby causing the bridge devices to build respective databases for use in forwarding the traffic,**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.C.2.v and VII.C.1.vi. EX1003, ¶128.

- vi. **[24e] the protocol processor being further adapted, responsive to a failure associated with the first link, to place the second downstream port in the active state and to send dummy traffic over the second link downstream to the further bridge device, so as to cause the further bridge device to modify its database responsive to the second downstream port being in the active state.**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.C.2.vi, VII.C.2.vii and VII.C.1.vi. EX1003, ¶129.

4. Dependent Claims 2 and 14

- i. **[2pre/14pre] A method/device according to claim 1/13,**

Gai renders obvious claim 1/13. *See* Sections VII.C.1/VII.C.2. EX1003, ¶130.

- ii. **[2a/14a] wherein [placing the second port in the blocking state comprises configuring]/[wherein the protocol processor is adapted to configure] the first and second [access] ports so that no loop is created in the network between the edge switch and the one or more access switches.**

Gai discloses and/or renders obvious this limitation. For example, Gai implements the STP, which ensures there is no loop in the network. EX1005, 5:36-39; 2:30-41. A POSITA would have understood that no loop is created in the network among any of the network switches. EX1003, ¶131.

5. Dependent Claim 4

- i. **[4pre] A method according to claim 1, and comprising,**

Gai renders obvious claim 1. *See* Section VII.C.1. EX1003, ¶132.

- ii. **[4a] responsive to placing the second port in the active state, updating the database of the edge switch so that entries in the database point to the second port instead of the first port.**

Gai discloses and/or renders obvious this limitation. For example, Gai discloses that a switch, e.g., the edge switch, discards old information in their databases after a port transitions to an active state. EX1005, 4:30-34. (“As ports transition between the blocked and forwarding states...To prevent switches from distributing messages based upon incorrect information, *switches* quickly age-out and *discard the ‘old’ information in their filtering databases.*”). EX1003, ¶133.

In Gai, “filtering database 244” includes rows that store destination port information that corresponds to particular network entities. EX1005, 8:32-39 (“[A] filtering database 244 configured to store address information corresponding to the entities of network 100. Specifically, filtering database 244 has a plurality of columns or cells, including a destination address cell, *a destination port cell*.... Each row or entry in the filtering database 244 preferably corresponds to a particular entity.”). EX1003, ¶134.

Gai also discloses that “[s]witch 114...informs the other devices of its new forwarding port by transmitting dummy multicast packets *through the new port*...” EX1005, 12:62-65. A POSITA would have understood that the contents, e.g., the port information, of the edge switch database would be updated to identify the new port for a particular entity, such that the dummy message(s) and/or communication messages would be transmitted from the appropriate port. EX1003, ¶135.

Accordingly, it would have been obvious to update the edge switch database to reflect the change in the active ports, such that when rapid reconfiguration entity 234 generates the dummy message(s) or when forwarding entity 238 forwards communication messages they would use the new port. For example, in the switch 124 database, the destination port information in the row that corresponds to switch 116 would be updated to reflect the port identified with the **red dot**.

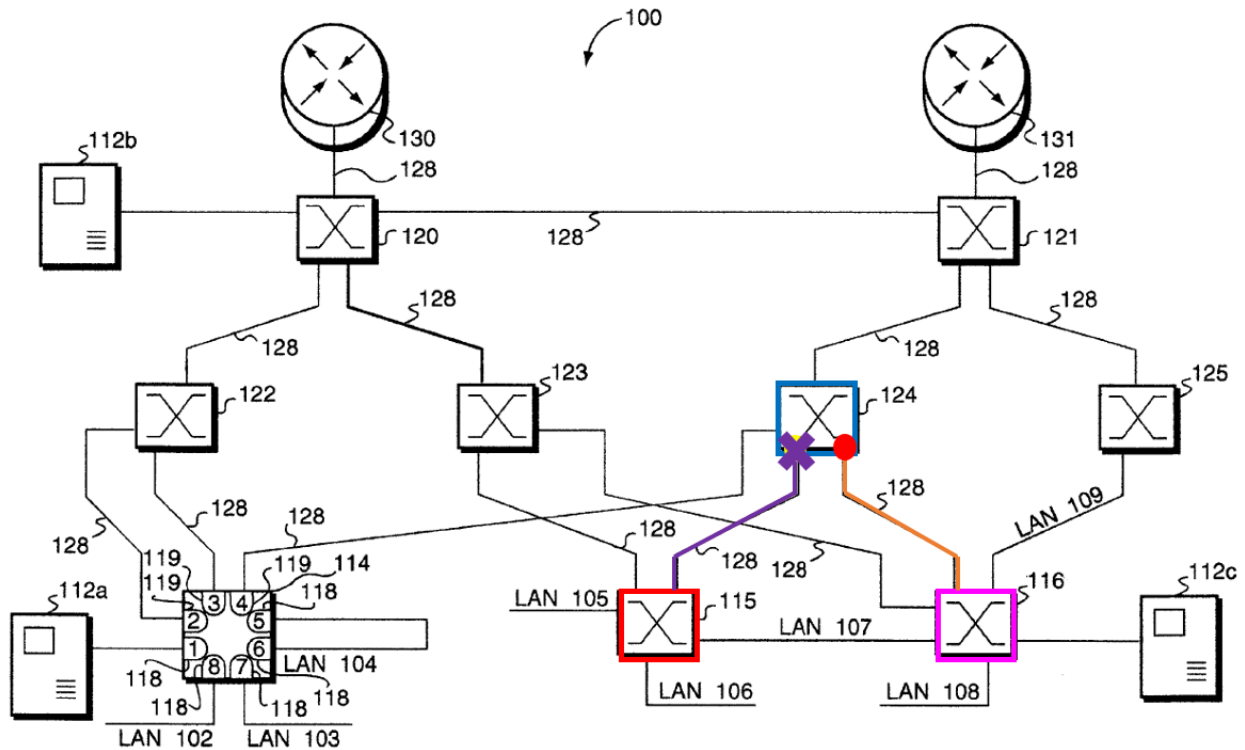


FIG. 1

EX1005, Figure 1. EX1003, ¶136.

6. Dependent Claims 5 and 17

i. [5pre/17pre] A method/device according to claim 4/16,

Gai renders obvious claims 4/16. *See* Sections VII.C.5/VII.C.12. EX1003, ¶137.

- ii. [5a/17a] wherein [updating the database comprises changing]/[the protocol processor is adapted to update] the entries in the database to point to the second [access] port before receiving the communication traffic from the one or more access switches on the second port.

Gai discloses and/or renders obvious this limitation. For example, Gai discloses that “[u]pon *transition* to the new forwarding port, the device begins transmitting ‘dummy’ multicast messages through the new port.” EX1005, 5:54-56. Gai also discloses that “[s]witch 114, via rapid reconfiguration entity 234, preferably informs the other devices of its new forwarding port by transmitting dummy multicast packets *through the new port...*” EX1005, 12:62-65. EX1003, ¶138.

Because Gai discloses that the dummy messages are transmitted upon transition to the new forward port, a POSITA would have understood that the dummy messages are transmitted before any communication traffic is received by the switch. As discussed in Section VII.C.4.ii, the database is updated before transmitting the dummy messages. Accordingly, a POSITA would have understood that the database entries would have been updated before receiving any communication traffic from the access switches. EX1003, ¶139.

7. Dependent Claims 6 and 18

i. [6pre/18pre] A method/device according to claim 1/13,

Gai renders obvious claim 1/13. *See* Sections VII.C.1/VII.C.2. EX1003, ¶140.

ii. [6a/18a] wherein [sending] the dummy traffic comprises [sending] data frames originating at the edge switch while having source addresses corresponding to network addresses of other entities in the network that are accessible to the access switches via the edge switch.

Gai discloses this limitation. For example, Gai discloses “[u]pon transition to the new forwarding port, *the device* begins transmitting ‘dummy’ multicast messages through the new port. These dummy multicast messages *carry the source address of each entity that is directly coupled to the device with the new active port or downstream thereof* (relative to the root) and are received by other devices in the computer network.” EX1005, 5:54-60; *see also*, EX1005, 13:14-33, 13:50-14:3; 11:16 (“As messages (i.e., data frames)...”). For example, upon transitioning to the new forwarding port, switch 124 would transmit a dummy message through the new port with the source address of server 112a (a device coupled downstream of switch 124). Server 112a is accessible to access switches 115 and 116 through switch 124.

reconfiguration entity 234 (FIG. 2) loads SA field 412 with the address of an entity directly coupled to switch 114. For example, in a first multicast message 400, *entity 234 loads the address of server 112a into the source address field 412.*"); 13:50-55 ("Rapid reconfiguration entity 234 similarly generates one or more separate multicast messages 400 for each remaining entity directly coupled to access switch 114 (i.e., entities or end stations on LANs 102-104). In each message 400, entity 234 *loads SA field 414 with the address of the corresponding entity.*"). EX1003, ¶144.

Gai's "filtering database 244 [is] configured to store **address information** corresponding to the entities of network 100. Specifically, filtering database 244 has a plurality of columns or cells, including a *destination address cell...*" and "[e]ach row or entry in the filtering database 244 preferably corresponds to a particular entity." EX1005, 8:31-39. Accordingly, a POSITA would have found it obvious to copy into the SA field of the dummy message the information from the "destination address cell," which corresponds to the "address information corresponding to" each entity of network 100. For example, a POSITA would have performed a look up function at the filtering database to retrieve the address information. *See also*, EX1005, 11:19-21 ("The forwarding entity 238...performs a look up function at the filtering database 244."). EX1003, ¶¶145-146.

9. Dependent Claims 9 and 21

i. [9pre/21pre] A method/device according to claim 6/18,

Gai renders obvious claim 6/18. *See* Section VII.C.7. EX1003, ¶147.

ii. [9a/21a] [wherein sending the data frames comprises incorporating]/[wherein the protocol processor is adapted to incorporate] a Virtual Local Area Network (VLAN) tag in a header of the data frames.

Gai discloses this limitation. For example, Gai discloses that “[r]eferring again to FIG. 4, in formulating dummy multicast messages following the detection of a change, access switches 114-116 *may append a VLAN tag 416 to message 400...*” EX1005, 17:32-39. A POSITA would have understood that backbone switches would also append a VLAN tag to frames/messages, because “an objective of [the] VLAN arrangement is *to allow all ports and entities of the network* having the same VLAN designation to interchange messages by associating a VLAN designation with each message. Thus, those entities having the same VLAN designation function as if they are all part of the same LAN.” EX1005, 16:17-22. Accordingly, when any switch sends a frame in a VLAN arrangement, the VLAN tag would be appended to the frame, such that the frame reaches the correct VLAN destination.

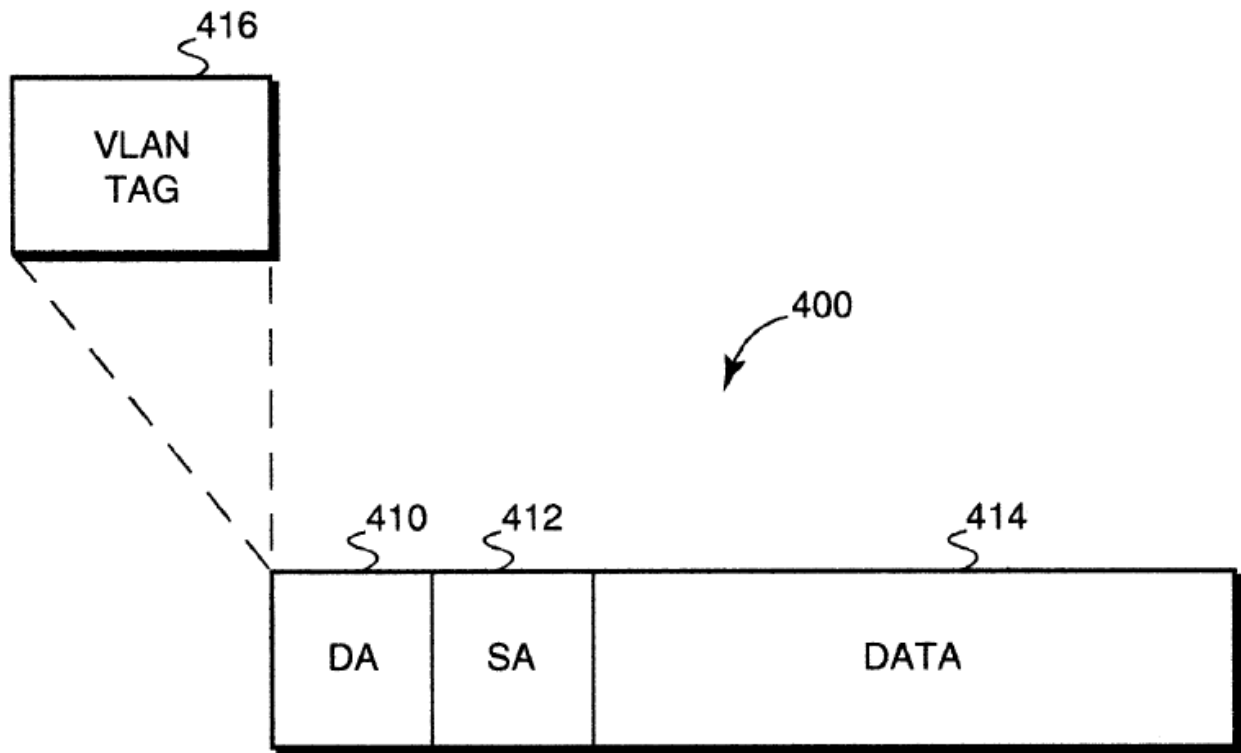


FIG. 4

EX1005, Figure 4; *see also*, EX1005, 15:52-16:27, 17:8-56. EX1003, ¶¶148-150.

10. Dependent Claims 10 and 22

- i. [10pre/22pre] A method/device according to claim 6/18,

Gai renders obvious claim 6/18. *See* Section VII.C.7. EX1003, ¶151.

- ii. **[10a/22a] [wherein sending the data frames comprises multicasting]/[wherein the protocol processor is adapted to multicast] the data frames to all of the one or more access switches.**

Gai discloses this limitation. For example, Gai discloses that “[s]witch 114, via rapid reconfiguration entity 234, preferably informs the other devices of its new forwarding port by transmitting *dummy multicast packets* through the new port,” (EX1005, 12:62-65) and that “[r]apid reconfiguration entity 234 preferably loads DA field 410 with a *multicast address that causes message 400 to be received by all devices* (e.g., access switches 114-116 and backbone switches 120–125) within the network 100.” EX1005, 13:9-13; *see also*, EX1005, 5:54-56, Abstract (“Following the transition to a new active port, *dummy multicast messages are transmitted*, each containing the source address of an entity directly coupled to the affected device or downstream thereof.”). EX1003, ¶152.

Gai also discloses that “[b]ackbone switch 124 distributes the multicast message 400 through all of its forwarding ports so that other devices, as necessary, may learn of the new forwarding port...at access switch 114.” EX1005, 13:45-49. EX1003, ¶153.

11. Dependent Claims 11 and 23

- i. **[11pre/23pre] A method/device according to claim 1/13,**

Gai renders obvious claim 1/13. *See* Sections VII.C.1/VII.C.2. EX1003, ¶154.

- ii. **[11a/23a] wherein [sending the dummy traffic comprises sending data frames downstream]/[the dummy traffic comprises data frames sent downstream] from the edge switch toward the access switches.**

Gai renders obvious this limitation for the reasons discussed in Sections VII.C.1.ii and VII.C.1.vi. EX1003, ¶155.

12. Dependent Claims 16 and 19

- i. **[16pre/19pre] A device according to claim 13/18, and**

Gai renders obvious claim 13/18. *See* Sections VII.C.2/VII.C.7. EX1003, ¶156.

- ii. **[16a/19a] comprising a memory,**

Gai discloses this limitation. For example, Gai discloses that “[s]witches typically learn which destination port to use in order to reach a particular entity by noting on which source port the last message originating from that entity was received. This information is then stored by each switch in *a block of memory* referred to as a filtering database.” EX1005, 1:50-54. Gai also discloses “that

memory...244 may be implemented as *content addressable memory (CAM) devices...*”. EX1005, 8:40-45; *see also*, EX1005, 8:31-40. EX1003, ¶¶157-158.

- iii. **[16b/9b] wherein the protocol processor is adapted to build an edge switch database in the memory responsive to the communication traffic prior to the failure associated with the first link[,] and,**

Gai discloses and/or renders obvious this limitation. For example, Gai discloses that “[s]witches typically learn which destination port to use in order to reach a particular entity by noting on which source port the last message originating from that entity was received. *This information is then stored by each switch in a block of memory referred to as a filtering database.*” EX1005, 1:50-54. Gai also discloses discarding this information, e.g., upon transitioning to a new port. EX1005, 4:30-34 (“As ports transition between the blocked and forwarding states, entities may appear to move from one port to another. To prevent switches from distributing messages based upon incorrect information, switches quickly age-out and *discard the ‘old’ information in their filtering databases.*”). Accordingly, Gai discloses that the switch databases are built prior to a transition to a new port, which can be, for example, as a result of a failure. *See e.g.*, EX1005, 5:44-46 (“Upon detection of a failure at the active forwarding port, the state of one of the back-up ports immediately transitions from blocked to forwarding...”). EX1003, ¶¶159-160.

Gai further explains:

As messages...are...received at the ports 230, they are passed to forwarding entity 238...The forwarding entity 238 first examines the destination address of the message and performs a look up function at the filtering database 244. *Assuming an entry is found* and a corresponding destination port...is identified, the message is switched out onto this port, assuming this port is in the forwarding state. If no entry is found..., the message is flooded out all ports 230 in the forwarding state, except the port 230 on which the message was received. The forwarding entity 238 next examines the source address of the message and performs another look-up at filtering database 244. *If no entry is found for the source address, a new entry is formed* and the source port...on which the message was received is entered as the destination port in the corresponding port column...

Should a change occur in the network 100, such as a failure..., the affected access switch will be able to rapidly reconfigure the network 100 without the significant interruption or message flooding experienced through conventional operation of the spanning tree algorithm.

EX1005, 11:16-41.

By quickly *discarding the contents of their filtering databases 244*, the possibility of data messages being misdirected or lost is significantly reduced....

EX1005, 15:18-25. EX1003, ¶161.

- iv. **[16c] responsive to placing the second access port in the active state, to update the edge switch database so that entries in the edge switch database point to the second access port instead of the first access port.**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.5.ii. EX1003, ¶162.

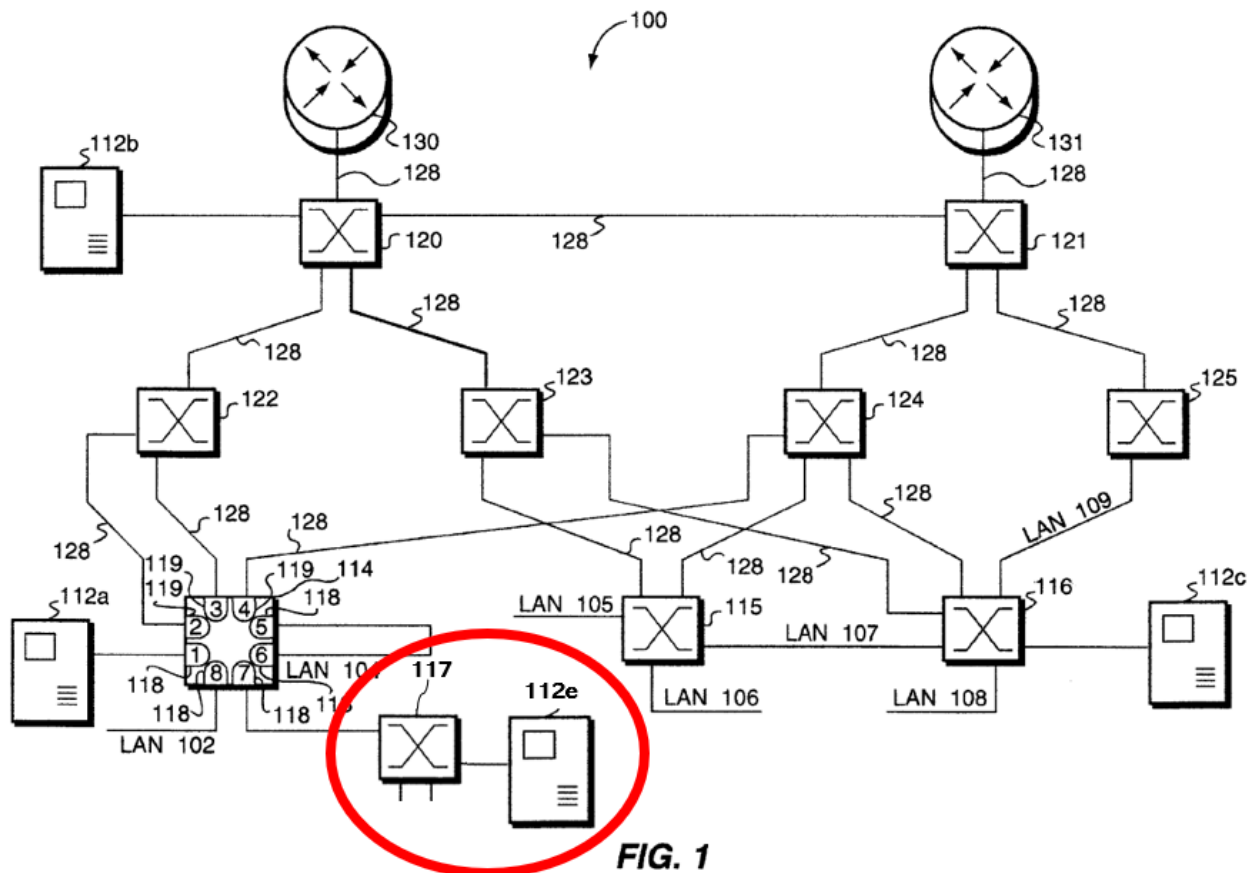
- v. **[19c] to copy the source addresses for the data frames from the edge switch database.**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.C.8.ii. EX1003, ¶163.

D. Ground 1B: Claims 1-2, 4-7, 9-14, 16-19, And 21-24 Are Obvious Over Gai (Upstream Embodiment)

At the outset, Applicant of the '986 Patent admitted that Gai discloses sending dummy packets *upstream* and distinguished the alleged '986 Patent invention as “sending *dummy traffic...downstream*” (EX1001, 12:4-5), which is “*in contrast* to the methods described in [Gai]...”. EX1001, 4:39-40. But not all claims require sending dummy packets downstream, and the Examiner appears to have overlooked this fact. Ground 1B explains how Gai’s teachings regarding sending dummy packets upstream render obvious the challenged claims. The analysis of Ground 1A that is not specifically directed to Gai’s downstream embodiment also applies here. EX1003, ¶164.

Gai's teachings apply to variations of the network of Figure 1. EX1005, 7:41-44 ("[T]he network 100 of FIG. 1 is meant for illustrative purposes only and that the present invention will operate with other network designs having possibly far more complex topologies."). Gai's upstream embodiment is discussed in connection with a modified Figure 1. For example, Gai discloses that port seven of switch 114 can be coupled to another switch. EX1005, 13:60-63 ("[I]f port number seven at switch 114 were *coupled to another access switch*, entity 234 would transmit multicast messages 400 carrying the source address of *entities directly coupled to this additional access switch* as well."). Such network is illustrated below:



EX1005, Figure 1 (modified to add switch 117 and server 112e). EX1003, ¶165.

In this exemplary network, **switch 114** is both an access switch and an **edge switch**, because it is a core switch (a switch that interconnect access switches 115-117, 120, and 125 into a WAN, e.g., by direct connection between core switch 114 and access switch 117) that is directly connected to an access switch (**switch 117**). EX1003, ¶166.

1. Independent Claims 1/12

i. [1pre] A method for network communication, comprising:

To the extent the preamble is limiting, Gai discloses and/or renders obvious the preamble. *See* Section VII.C.1.i. EX1003, ¶167.

ii. [1a] coupling an edge switch in a network to one or more access switches via at least first and second redundant links, respectively connected to first and second ports of the edge switch;

[12a] coupling a first bridge in a network to a second bridge located downstream of the first bridge, by connecting at least first and second redundant links to respective first and second ports of the first bridge so as to communicate with the second bridge;

Gai discloses and/or renders obvious limitations [1a/12a]. At the outset, a “switch” is one particular type of a “bridge.” *See, e.g.*, EX1001, 1:26; 4:50-54. EX1003, ¶168.

Gai discloses an **edge switch** and one or more **access switches**, as the '986 Patent admits. *See* EX1001, 3:16-24. Indeed, as shown below, Gai's **switch 114** meets the claimed **edge switch/first bridge** and Gai's **access switch 120** meets the claimed **one or more access switches/second bridge**. EX1003, ¶¶169-170.

Switch 114, as shown in the two examples below, meets the claimed **edge switch** because **edge switches** are “[c]ore switches that are directly connected to **access switches**.” EX1001, 1:36-38. *First*, **switch 114** is directly connected to **access switch 117**. *Second*, **switch 114** is a core switch because it “interconnect[s] **access switch [117]** into a [WAN], by direct connection between the core switch and access switches and/or by connection to other core switches [such as switch 124].” EX1001, 1:33-36; *see also* EX1005, 1:31-34.

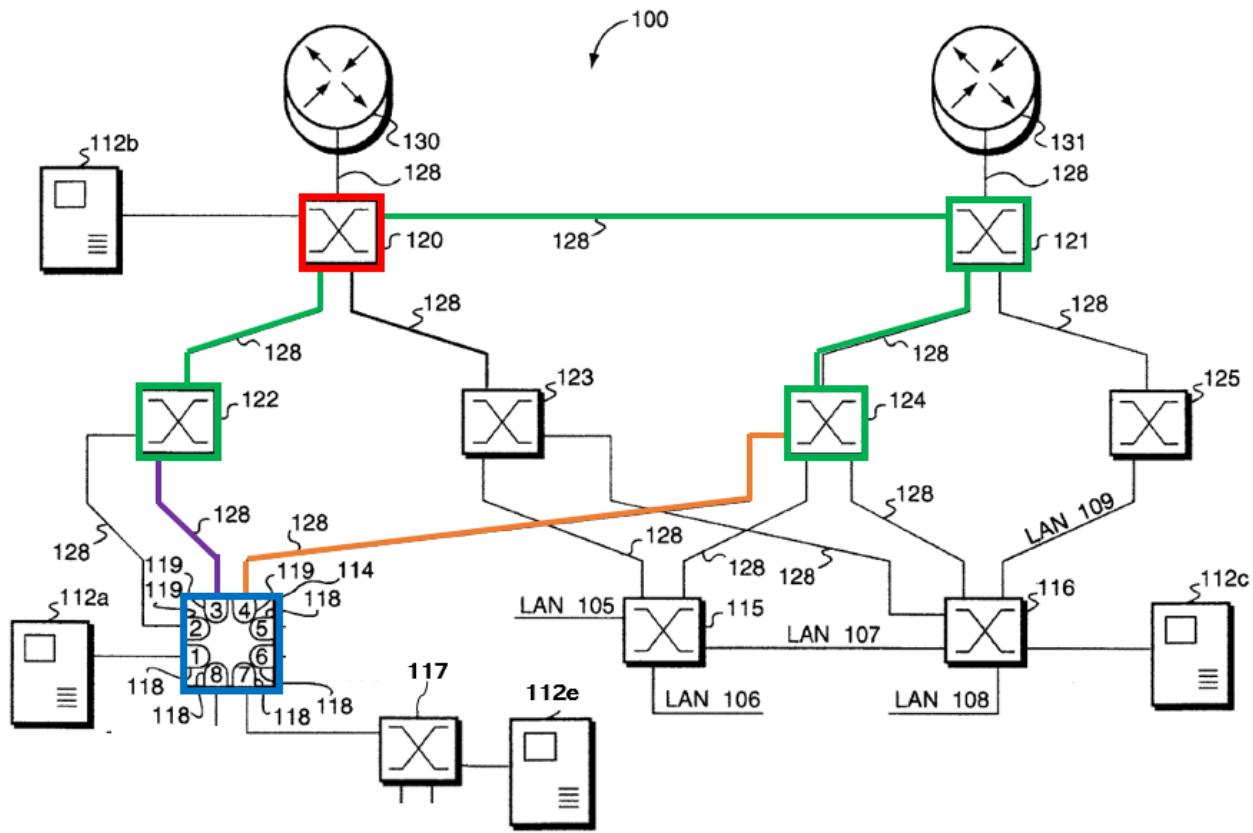


FIG. 1

EX1005, Figure 1 (modified). EX1003, ¶171.

Gai also discloses coupling the **edge switch** to the one or more **access switches** via **at least first** and **second** redundant links, respectively connected to first (port 3) and second (port 2/4) ports of the **edge switch**. EX1003, ¶172.

Limitation 12a further specifies that the “**second bridge** [is] located downstream of the **first bridge**,” i.e., away from the root switch. See EX1001, 3:30-31. But, as discussed in Section III.A.2, any switch can be the root switch, e.g., the switch with the lowest switch identifier. EX1005, 2:42-47; 10:29-32; 3:17-39. Accordingly, it would have been obvious that **second bridge 120** would be located

downstream of *first bridge 114*, if switch 117 was the root, because, for example, it was the switch with the lowest switch identifier.

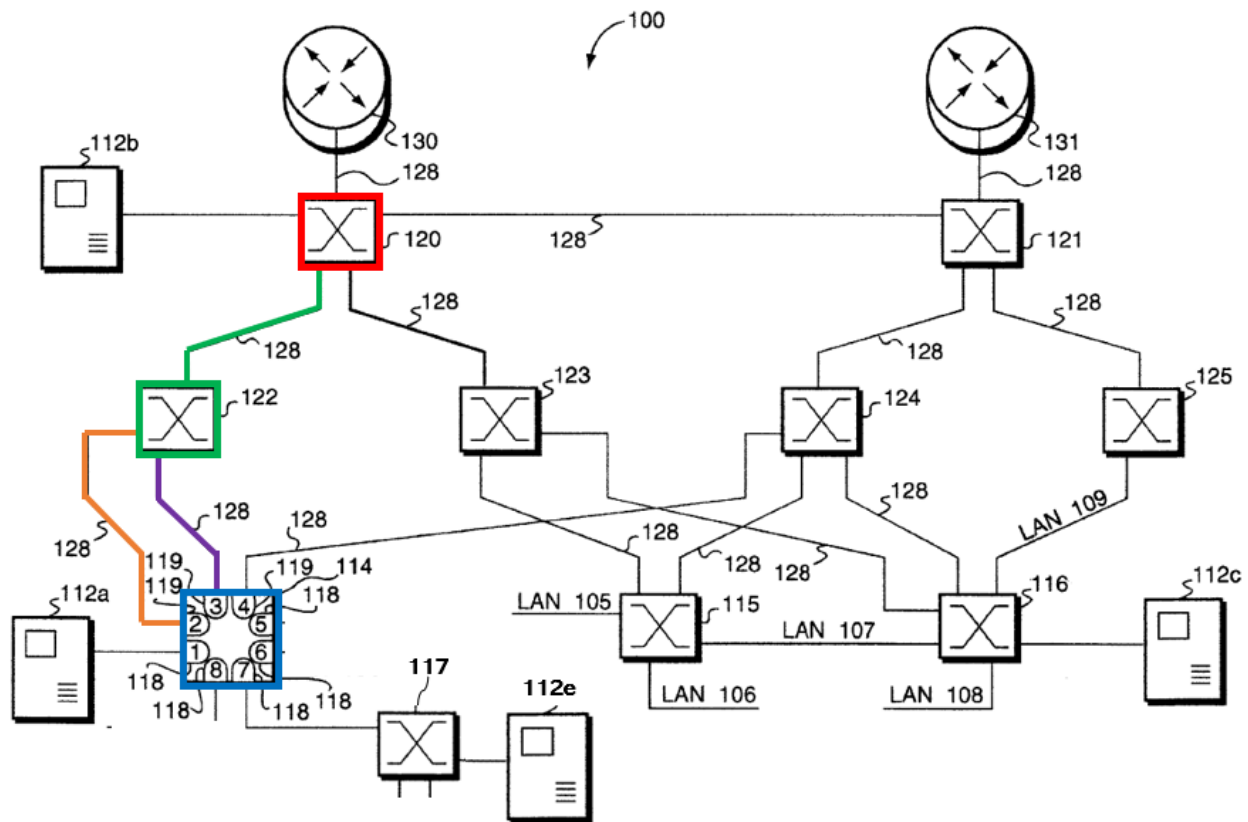


FIG. 1

EX1005, Figure 1 (modified).

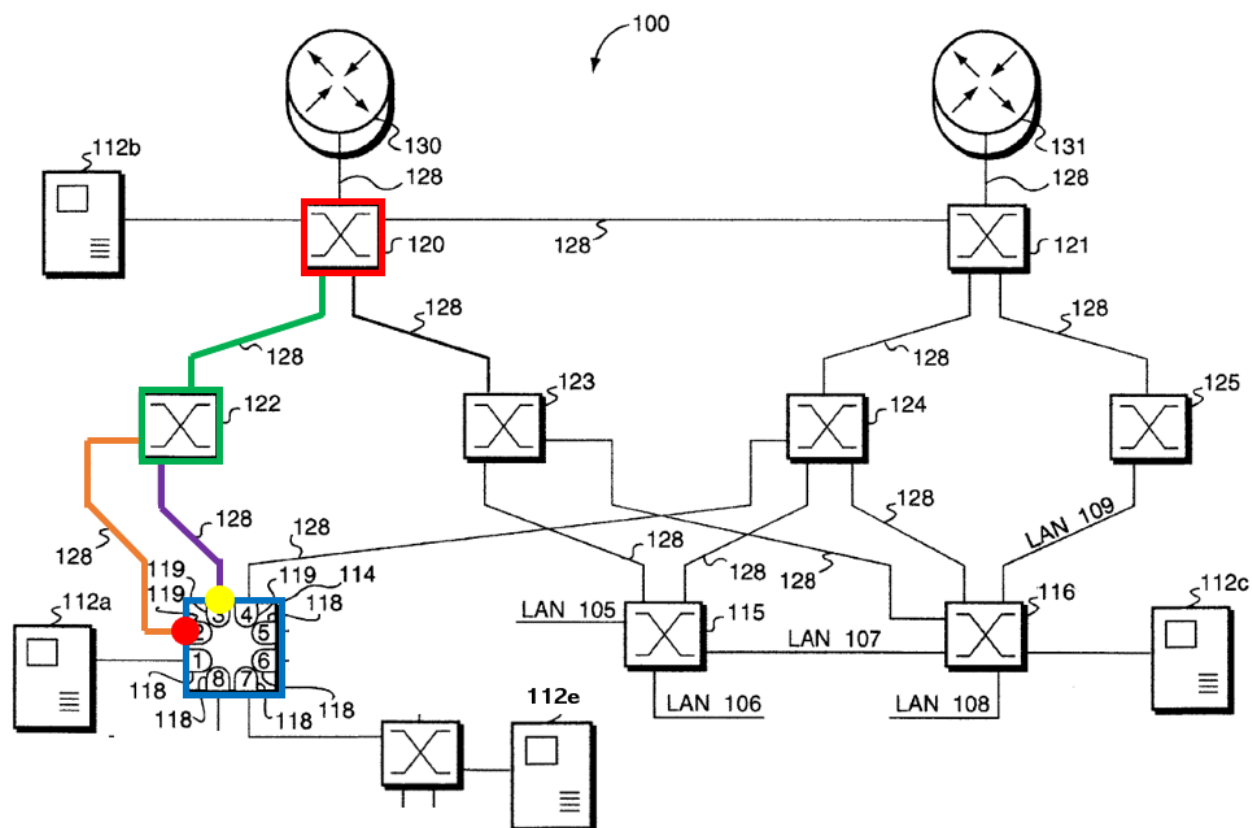


FIG. 1

EX1005, Figure 1 (modified).

- v. [1d/12d] responsive to a failure associated with the first link, placing the second port in the active state and the first port in the blocking state; and

Gai discloses and/or renders obvious this limitation. At the outset, the '986 Patent admits that Gai discloses this limitation. EX1001, 3:26-29. Indeed, Gai discloses that “[u]pon detection of a failure at the active forwarding port [indicated with purple X], the state of one of the back-up ports [in red dot] immediately transitions from blocked to forwarding, thereby becoming the new active port for the device.” EX1005, 5:44-47; *see also*, EX1005, Abstract, 5:20-23, 11:36-41.

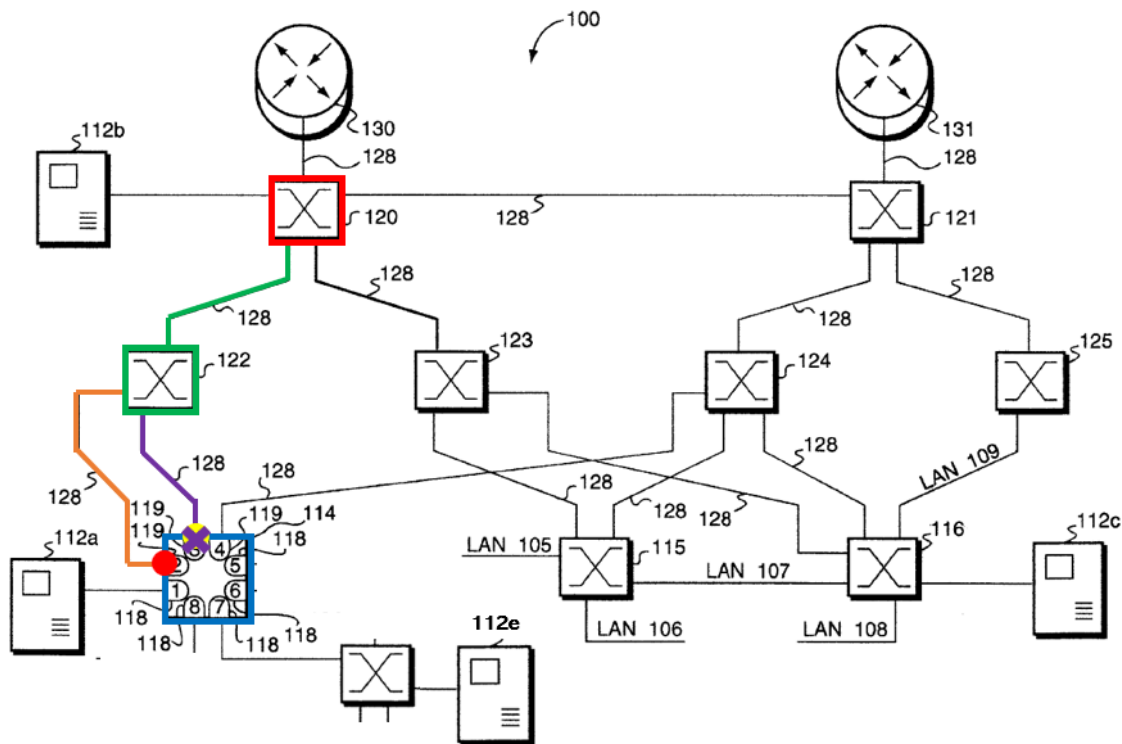


FIG. 1

EX1005, Figure 1 (modified).

then directs the spanning tree state machine engine 236 *to immediately transition the selected back-up port (e.g., port number four) to the forwarding state.*”).

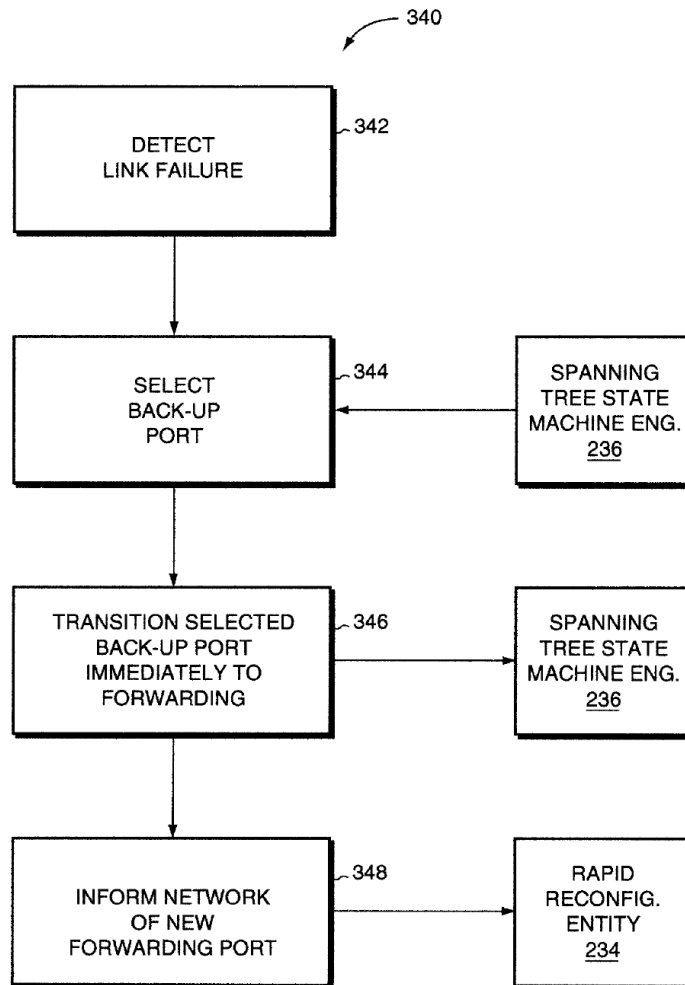


FIG. 3D

EX1005, Figure 3D; *see also*, EX1005, 12:29-31. EX1003, ¶183.

- vi. [1e] sending dummy traffic from the **edge switch** over the second link to the one or more **access switches**, so as to cause each of the one or more **access switches** to modify its respective database responsive to the second port being in the active state.

[12e] sending dummy traffic from the **first bridge** over the second link downstream to the **second bridge**, so as to cause the **second bridge** to modify its respective database responsive to the second port being in the active state.

Gai discloses and/or renders obvious this limitation. As discussed in Section VII.B, the '986 Patent admits that Gai discloses sending dummy traffic from a switch such that other switches learn of the new active port. *See* EX1001, 3:29-35. This admission is explicitly directed to an upstream transmission of dummy messages—which satisfies claim 1. However, Gai also renders obvious downstream transmission of dummy messages, as discussed below. EX1003, ¶184.

First, Gai discloses sending dummy traffic from the **edge switch** over the second link to the one or more **access switches**. Gai discloses that “the transition at access switch 114 from initial root port number three to back-up root port number four [or two] may cause entities to appear to ‘move’ relative to other devices. Thus, *the identity of the new root port (e.g., port number four) must be propagated to the other devices*, such as access switches 115-116 and backbone switches 120-125 to prevent messages from being lost or misdirected.” EX1005, 12:56-62; *see also*, EX1005, 12:28-31. Gai discloses transmitting dummy messages through the newly-

activated port (port 2/port 4) of switch 114 to inform the other devices of the newly-activated port. EX1005, 12:62-66 (“Switch 114, via rapid reconfiguration entity 234, preferably informs the other devices of its new forwarding port ***by transmitting dummy multicast packets through the new port***, [block 348].”); 11:48-52 (“If the link 128 coupled to the port number three at switch 114 fails, either one of these two other ports (i.e., port numbers two or four) will immediately transition to the forwarding state and begin receiving and sending messages.”); *see also*, EX1005, 5:54-56, 12:66-13:49. EX1003, ¶¶185-186.

Gai further discloses that “[r]apid reconfiguration entity 234 similarly generates one or more separate multicast messages 400 for each remaining entity directly coupled to access switch 114...”. EX1005, 13:50-53; *see also*, EX1005, 13:53-14:3.

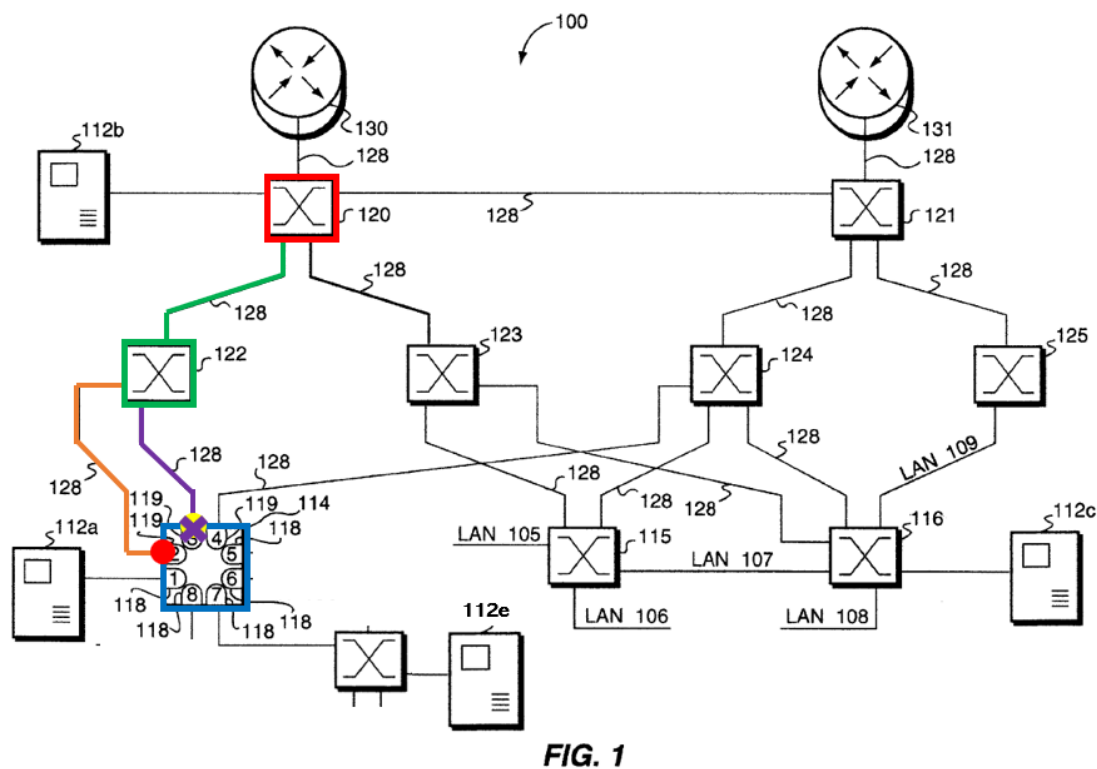


FIG. 1

EX1005, Figure 1 (modified).

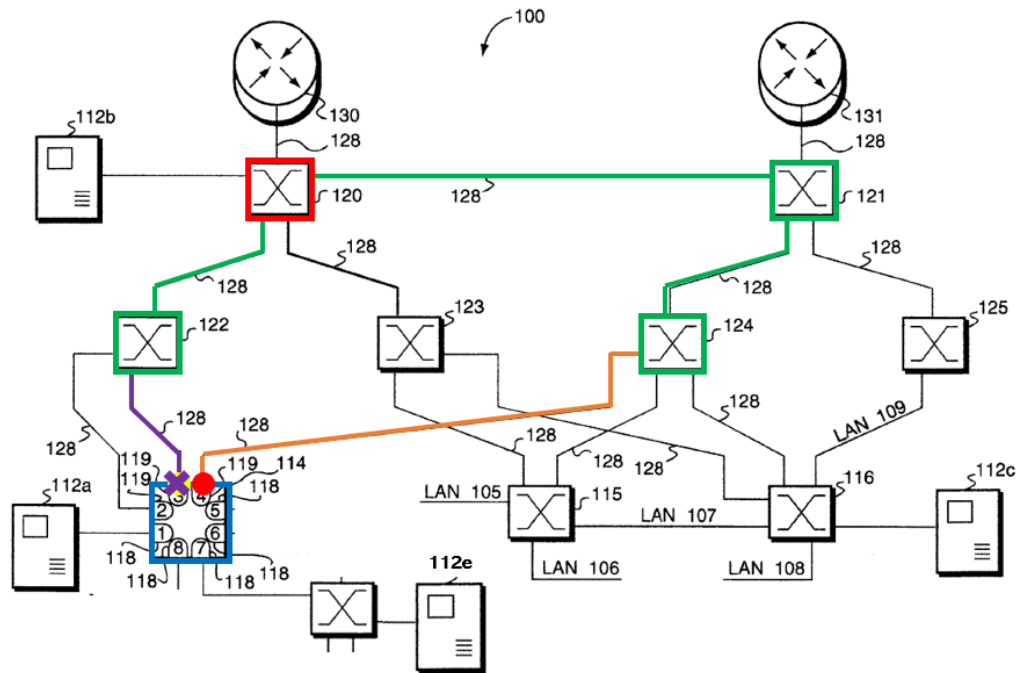


FIG. 1

EX1005, Figure 1 (modified). EX1003, ¶187.

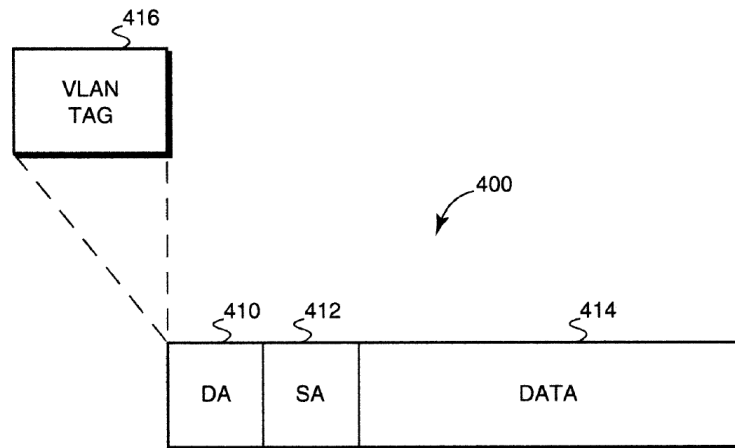


FIG. 4

EX1005, Figure 4; *see also*, EX1005, 12:66-13:1 (“FIG. 4 is a highly schematic illustration of a *dummy multicast message* 400...”). EX1003, ¶188.

Second, Gai discloses “so as to cause each of the one or more *access switches* to modify its respective database responsive to the second port being in the active state,” for the reasons explained in Section VII.C.1.vi. For example, when access switch 120 receives a dummy message from switch 121 with a source address corresponding to server 112a, it would update its corresponding database, such that the next time switch 120 transmits a frame addressed to server 112a, it would use the appropriate port, e.g., the port to send the frame towards switch 121. EX1003, ¶189.

Limitation 12e adds that the “dummy traffic” is sent downstream (away from the root switch). *See* EX1001, 3:30-31. Gai renders obvious this additional limitation. For example, Figure 1 “assume[s] that backbone *switch 121*...is the root...” EX1005, 14:16-17. But, as discussed in Section III.A.2, any switch in the network can be the root, e.g., the switch with the lowest switch identifier. EX1005, 2:42-47; 10:29-33; 3:17-39. Accordingly, it would have been obvious that *first bridge 114* would send dummy traffic downstream through the new active ports (indicated by the red dot) over *link 128*, away from the root switch and towards *second bridge 120*, if switch 117 was the root, because, e.g., it was the switch with the lowest switch identifier.

2. Independent Claim 13

i. [13pre] A **network switching device**, comprising:

To the extent the preamble is limiting, Gai discloses and/or renders obvious the preamble. For example, Gai's Figure 1 discloses various **network switching devices**, e.g., **switch 114**.

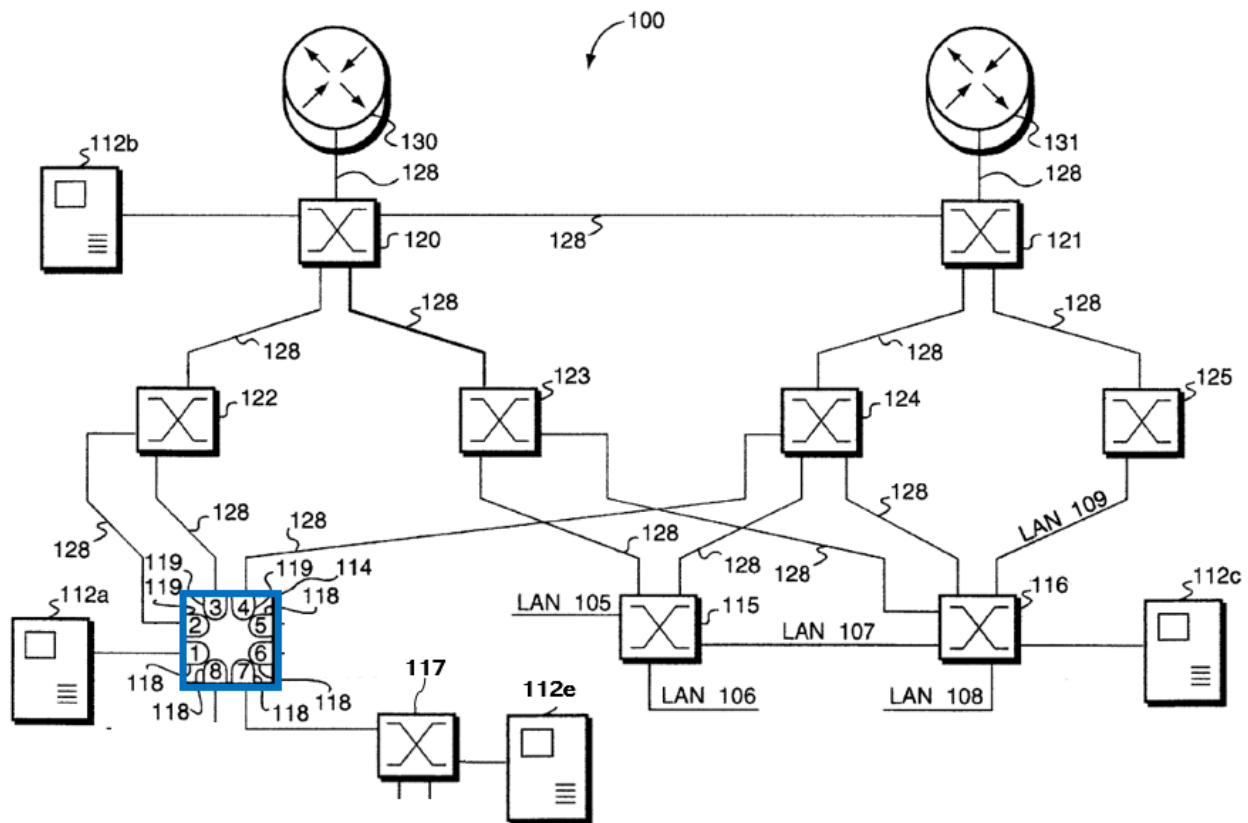


FIG. 1

EX1005, Figure 1. EX1003, ¶191.

- ii. [13a] a network port, adapted to be coupled over a network trunk link to a core switch in a network so as to exchange communication traffic with the core switch;

Gai discloses and/or renders obvious this limitation. For example, [switch 114](#) has network port seven coupled to switch 117 over link 128 identified with a [green arrow](#), and meets the claimed “trunk link.”

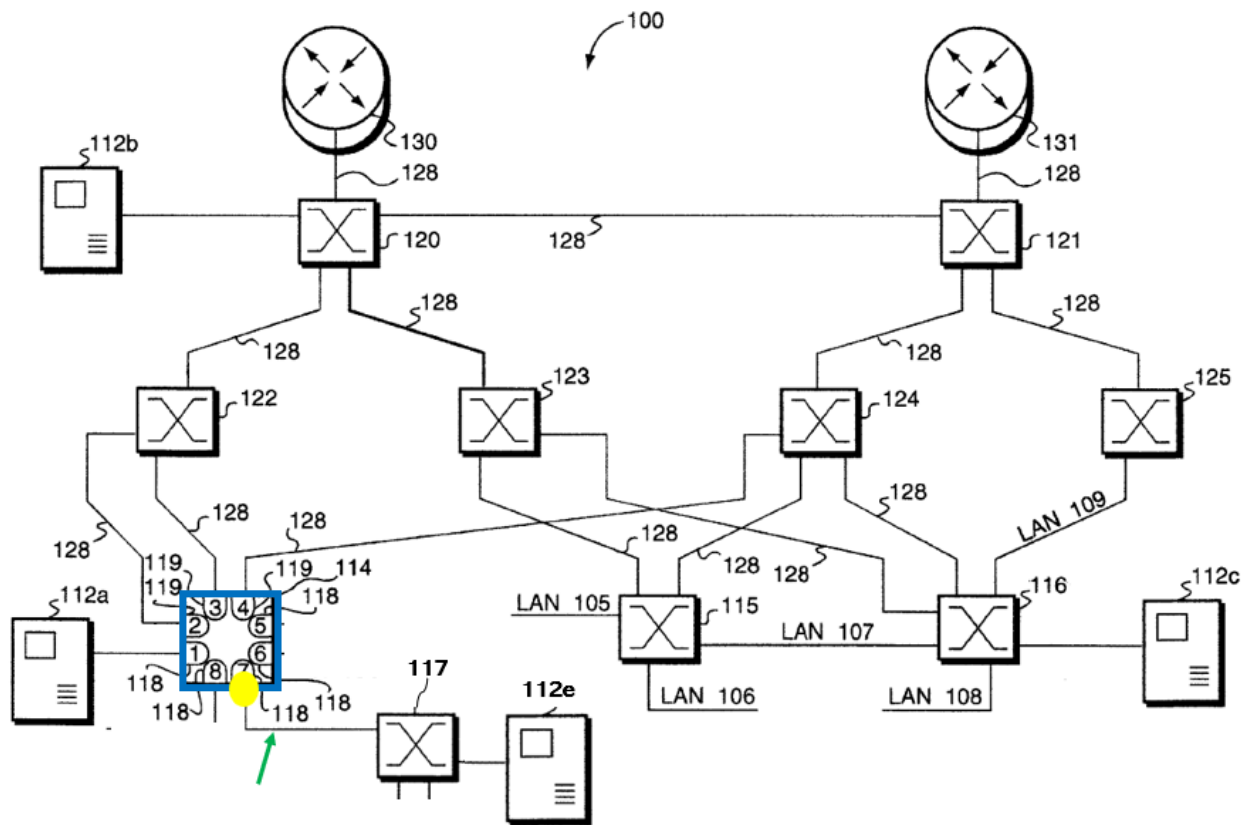


FIG. 1

EX1005, Figure 1 (modified). EX1003, ¶192.

Moreover, a POSITA would have also understood that network port seven of [switch 114](#) exchanges communication traffic with switch 117 because it is directly

connected—through the link—to switch 117. Switch 117 is shown as an access switch in modified Figure 1 above, but a POSITA would have understood that switch 117 would satisfy a “core” switch, if, for example, one of its ports was connected to another switch. EX1003, ¶193.

**iii. [13b] first and second access ports, ...network users;
and**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.1.iii and VII.C.2.iii. EX1003, ¶194.

**iv. [13c] a protocol processor, adapted to ...a blocking
state,**

Gai discloses and/or renders obvious this limitation. *See* Section VII.C.2.iv. EX1003, ¶195.

v. [13d] so that the communication traffic ...traffic,

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.D.1.iv. EX1003, ¶196.

vi. [13e] the protocol processor ...blocking state and

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.1.v and VII.C.2.vi. EX1003, ¶197.

- vii. **[13f] [the protocol processor being further adapted, responsive to a failure associated with the first link] to send ...in the active state.**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.1.vi and VII.C.2.vii. EX1003, ¶198.

3. Independent Claim 24

- i. **[24pre] A bridge device ...the device comprising:**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.2.i, VII.D.1.ii, VII.D.1.vi. EX1003, ¶199.

- ii. **[24a] a designated port, adapted to be coupled over an upstream link in the network so as to communicate with the root;**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.D.2.ii. Additionally, the link connected to port 7 of switch 114 would be an upstream link, if switch 117 was the root, because, e.g., it was the switch with the lowest switch identifier, as discussed in Section VII.D.1.ii. EX1003, ¶200.

- iii. **[24b] first and second downstream ports, ...the spanning tree root; and**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.1.ii and VII.D.1.vi. EX1003, ¶201.

- iv. **[24c] a protocol processor, ...in a blocking state,**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.2.iv and VII.D.1.vi. EX1003, ¶202.

- v. **[24d] so that communication traffic ... in forwarding the traffic,**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.2.v and VII.D.1.vi. EX1003, ¶203.

- vi. **[24e] the protocol processor being further adapted, ... in the active state.**

Gai discloses and/or renders obvious this limitation for the reasons discussed in Sections VII.D.2.vi, VII.D.2.vii, and VII.D.1.vi. EX1003, ¶204.

4. Dependent Claims 2/14

- i. **[2pre/14pre] A method/device according to claim 1/13,**

Gai renders obvious claim 1/13. *See* Sections VII.D.1/VII.D.2. EX1003, ¶205.

- ii. **[2a/14a] wherein...access switches.**

Gai discloses and/or renders obvious this limitation. *See* Section VII.C.4.ii. EX1003, ¶206.

5. Dependent Claim 4

- i. **[4pre] A method according to claim 1, and comprising,**

Gai renders obvious claim 1. *See* Section VII.D.1. EX1003, ¶207.

- ii. [4a] responsive to placing the second port in the active state, updating the database of the edge switch so that entries in the database point to the second port instead of the first port.**

Gai discloses and/or renders obvious this limitation. For example, Gai discloses this limitation for the reasons discussed in VII.C.4.ii. EX1003, ¶208.

Alternatively, it would have been obvious to update the edge switch database to reflect the change in the active ports, such that when rapid reconfiguration entity 234 generates the dummy message(s) or when forwarding entity 238 forwards communication messages they would use the new port. For example, in the database of switches 122 and 124, the destination port cell information in the row that corresponds to switch 114 would be updated to reflect the ports two and four, respectively.

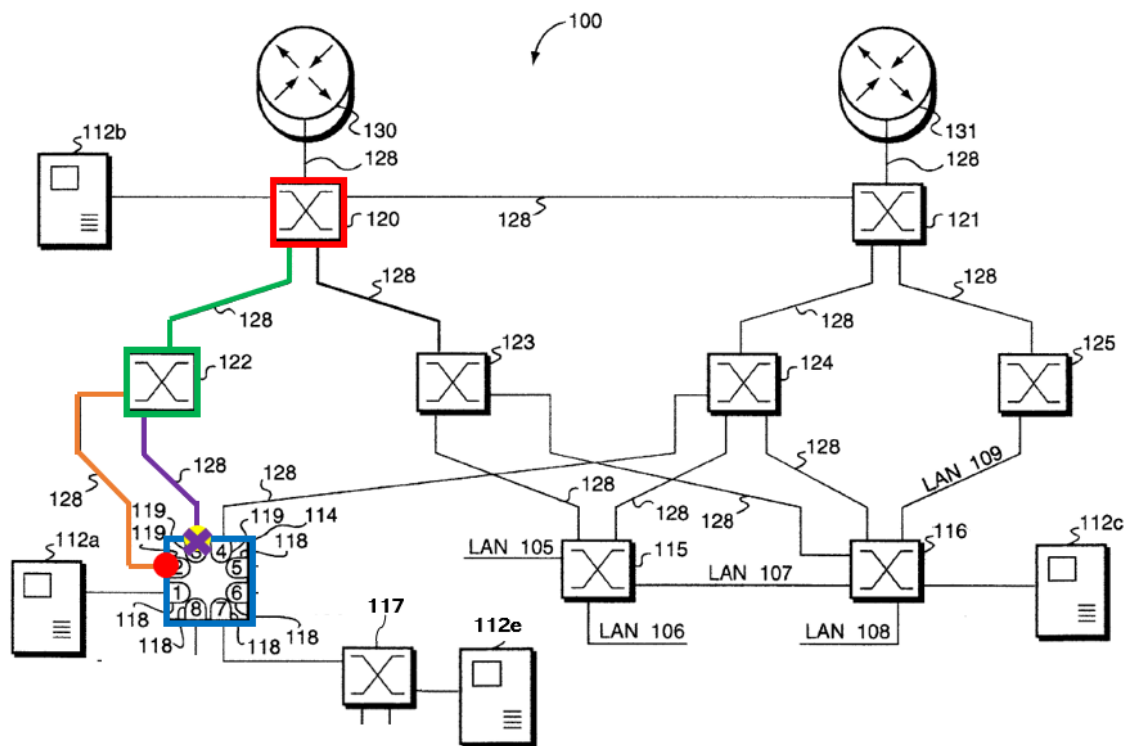


FIG. 1

EX1005, Figure 1 (modified).

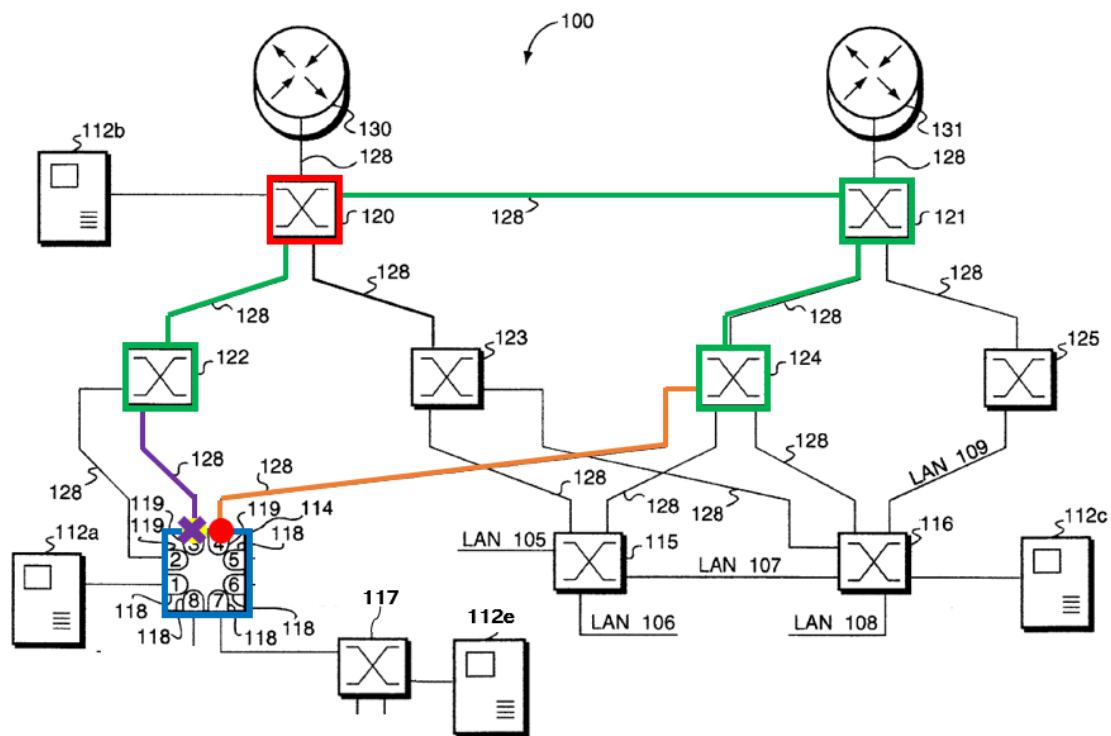


FIG. 1

EX1005, Figure 1 (modified). EX1003, ¶209.

6. Dependent Claims 5/17

i. [5pre/17pre] A method/device according to claim 4/16,

Gai renders obvious claim 4. *See* Section VII.D.5. EX1003, ¶210.

ii. [5a/17a] wherein...the second port.

Gai discloses and/or renders obvious this limitation. *See* Section VII.C.6.ii.

EX1003, ¶211.

7. Dependent Claims 6/18

i. [6pre/18pre] A method/device according to claim 1/13,

Gai renders obvious claim 1/13. *See* Sections VII.C.1/VII.C.2. EX1003, ¶212.

ii. [6a/18a] wherein...via the edge switch.

Gai discloses this limitation. *See* Section VII.C.7.ii. EX1003, ¶213.

8. Dependent Claim 7

i. [7pre] A method according to claim 6,

Gai renders obvious claim 6. *See* Section VII.C.7. EX1003, ¶214.

ii. [7a] wherein...the edge switch.

Gai renders obvious this limitation. *See* Section VII.C.8.ii. EX1003, ¶215.

9. Dependent Claims 9/21

- i. [9pre/21pre] A method/device according to claim 6/18,**

Gai renders obvious claim 6/18. *See* Section VII.D.7. EX1003, ¶216.

- ii. [9a/21a] wherein...the data frames.**

Gai discloses this limitation. *See* Section VII.C.9.ii. EX1003, ¶217.

10. Dependent Claims 10/22

- i. [10pre/22pre] A method/device according to claim 6/18,**

Gai renders obvious claim 6/18. *See* Section VII.D.7. EX1003, ¶218.

- ii. [10a/22a] wherein...access switches.**

Gai discloses this limitation. *See* Section VII.C.10.ii. EX1003, ¶219.

11. Dependent Claims 11/23

- i. [11pre/23pre] A method/device according to claim 1/13,**

Gai renders obvious claim 1/13. *See* Sections VII.C.1/VII.C.2. EX1003, ¶220.

- ii. [11a/23a] wherein...the access switches.**

Gai renders obvious this limitation for the reasons discussed in Sections VII.D.1.ii and VII.D.1.vi. EX1003, ¶221.

12. Dependent Claims 16/19

i. [16pre/19pre] A device according to claim 13/18, and

Gai renders obvious claim 13/18. *See* Sections VII.D.2/VII.D.7. EX1003, ¶222.

ii. [16a/19a] comprising a memory,

Gai discloses this limitation. *See* Section VII.C.8.ii. EX1003, ¶223.

iii. [16b/9b] wherein...first link[,] and,

Gai discloses this limitation. *See* Section VII.C.12.iii. EX1003, ¶224.

iv. [16c] responsive...the first access port.

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.D.4.ii. EX1003, ¶225.

v. [19c] to copy...edge switch database.

Gai discloses and/or renders obvious this limitation for the reasons discussed in Section VII.D.8.ii. EX1003, ¶226.

VIII. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

A. Discretionary Denial Under *Fintiv* Is Not Appropriate

The six factors considered for §314 denial favor institution. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential).

1. No evidence regarding a stay

No motion to stay has been filed, so this factor is neutral. *Sand Revolution II LLC v. Continental Intermodal Group – Trucking LLC*, IPR2019-01393, Paper 24, 7 (June 16, 2020) (informative); *see also Dish Network L.L.C. v. Broadband iTV, Inc.*, IPR2020-01359, Paper 15, 11 (Feb. 12, 2021) (“It would be improper to speculate, at this stage, what the Texas court might do regarding a motion to stay...”).

2. Parallel proceeding trial date

The Docket Control Order sets December 4, 2024, as the date for the claim construction hearing, and June 2, 2025, as the trial date. EX1008, 1, 4. The expected date for a FWD is August 2025.

This factor weighs against discretionary denial because the Petitioner has worked expeditiously to prepare and file this petition approximately one month after receiving Patent Owner’s infringement contentions on December 22, 2023. EX1015, 1; *Fintiv*, Paper 11, 11–12.

Furthermore, while trial is currently scheduled to begin in June 2025, “scheduled trial dates are often subject to change.” *In re Apple Inc.*, 979 F.3d 1332, 1344, n.5 (Fed. Cir. 2020). In *Sand*, the “relative close proximity” of a trial date scheduled five months before the FWD deadline “weigh[ed] marginally in favor of not exercising discretion to deny institution” given the district court’s own

recognition of a “continuing degree of...uncertainty of the court’s schedule.” *Sand*, 8-10. Here, even if the EDTX trial commences in June 2025, the FWD in this case is expected just two months later (August 2025), which is well within the 5-month time period in *Sand*.

Thus, this factor weighs against discretionary denial.

3. Investment in the parallel proceeding

The co-pending litigation is in its early stages. The investment in it has been minimal. Fact discovery and expert discovery are not set to close until January 13, 2025 and February 18, 2025, respectively. EX1008, 3. The Markman hearing date is set for December 4, 2024. EX1008, 4. Thus, this factor weighs against discretionary denial. *PEAG LLC v. Varta Microbattery GmbH*, IPR2020-01214, Paper 8, 17 (Jan. 6, 2021).

Under *Fintiv*, Petitioner’s prompt filing “weigh[s] against exercising the authority to deny institution.” *Fintiv*, Paper 11, 11 (“If the evidence shows that the petitioner filed the petition expeditiously, such as promptly after becoming aware of the claims being asserted, this fact has weighed against exercising the authority to deny institution under NHK”). This factor favors institution.

4. Overlapping issues with the parallel proceeding

Petitioner stipulates that if the IPR is instituted, Petitioner will not pursue the same ground in the district court litigation. *Sand*, 7.

5. Identity of parties

Petitioner is a defendant in the litigation. That is true of most Petitioners in IPR proceedings. Accordingly, this factor should not be a basis for denying institution.

6. Other circumstances

“[T]he PTAB will not deny institution...under *Fintiv* (i) when a petition presents compelling evidence of unpatentability.” EX1009, 9. Here, the evidence of unpatentability is compelling, and thus the PTAB should not deny institution under *Fintiv*.

B. Discretionary Denial Under 35 U.S.C. § 325(d) Is Not Appropriate Because The Examiner’s Material Errors Warrant Reaching The Merits Of This Petition

Section 325(d) does not apply here because the Examiner erred by not considering the relevant teachings in Gai. In deciding whether to exercise discretion to deny institution under §325(d), the Board determines first “whether the same or substantially the same art previously was presented to the Office or whether the same or substantially the same arguments previously were presented to the Office,” and second, “if either condition of first part of the framework is satisfied, whether the petitioner has demonstrated that the Office erred in a manner material to the patentability of challenged claims.” *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6, 8 (Feb. 13, 2020)

(precedential) (“*Advanced Bionics*”). The Board considers the factors set forth in *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17–18 (Dec. 15, 2017) (“*Becton, Dickinson*”). Specifically, the Board applies *Becton, Dickinson* factors (a), (b), and (d) to determine whether the same or substantially the same art or arguments previously were presented to the Office, and *Becton, Dickinson* factors (c), (e), and (f) to determine whether Petitioner has demonstrated a material error by the Office. *Advanced Bionics*, 10.

Here, Gai was presented to the Office in an IDS (EX1002, 46) and was discussed in the background section of the ’986 Patent. *See* EX1001, 3:15-41; 4:38-43. However, as explained below, Section 325(d) does not apply because the Examiner erred by not considering the relevant teachings in Gai.

1. *Advanced Bionics* Step Two: The Examiner Erred By Not Considering Gai And Relevant Teachings Therein

In the second step of the *Advanced Bionics* framework the Board determines whether Petitioner has demonstrated a material error by the Office, by examining three *Becton, Dickinson* factors: “(c) the extent to which the asserted art was evaluated during examination, including whether the prior art was the basis for rejection;” “(e) whether Petitioner has pointed out sufficiently how the Examiner erred in its evaluation of the asserted prior art; and (f) the extent to which additional

evidence and facts presented in the Petition warrant reconsideration of the prior art or arguments.” *Becton, Dickinson*, 17–18; *see Advanced Bionics*, 10.

Under factor (c), a reference previously presented to the Office in an IDS is not sufficient to satisfy *Advanced Bionics*. *Becton, Dickinson*, 23. Under factor (e) “[a]n example of a material error may include misapprehending or overlooking specific teachings of the relevant prior art where those teachings impact patentability of the challenged claims.” *Advanced Bionics*, 3. Here, the Petition explained how the Examiner erred in two separate ways. *First*, regarding Gai’s “upstream” embodiment, the Examiner either misapprehended or overlooked that Gai discloses sending dummy traffic upstream *and* that claim 1 covers sending dummy traffic upstream. For example, claim 11, which depends from claim 1, recites “sending the dummy traffic comprises *sending data frames downstream* from the edge switch toward the access switches.” EX1001, 11:53-55. Accordingly, the scope of claim 1 should cover both sending downstream and upstream data frames, since upstream transmission is the only alternative to downstream transmission.

Second, the Examiner either misapprehended or overlooked that Gai discloses and/or renders obvious sending dummy traffic downstream, as explained in Section VII.C.1.vi in connection with the example below.

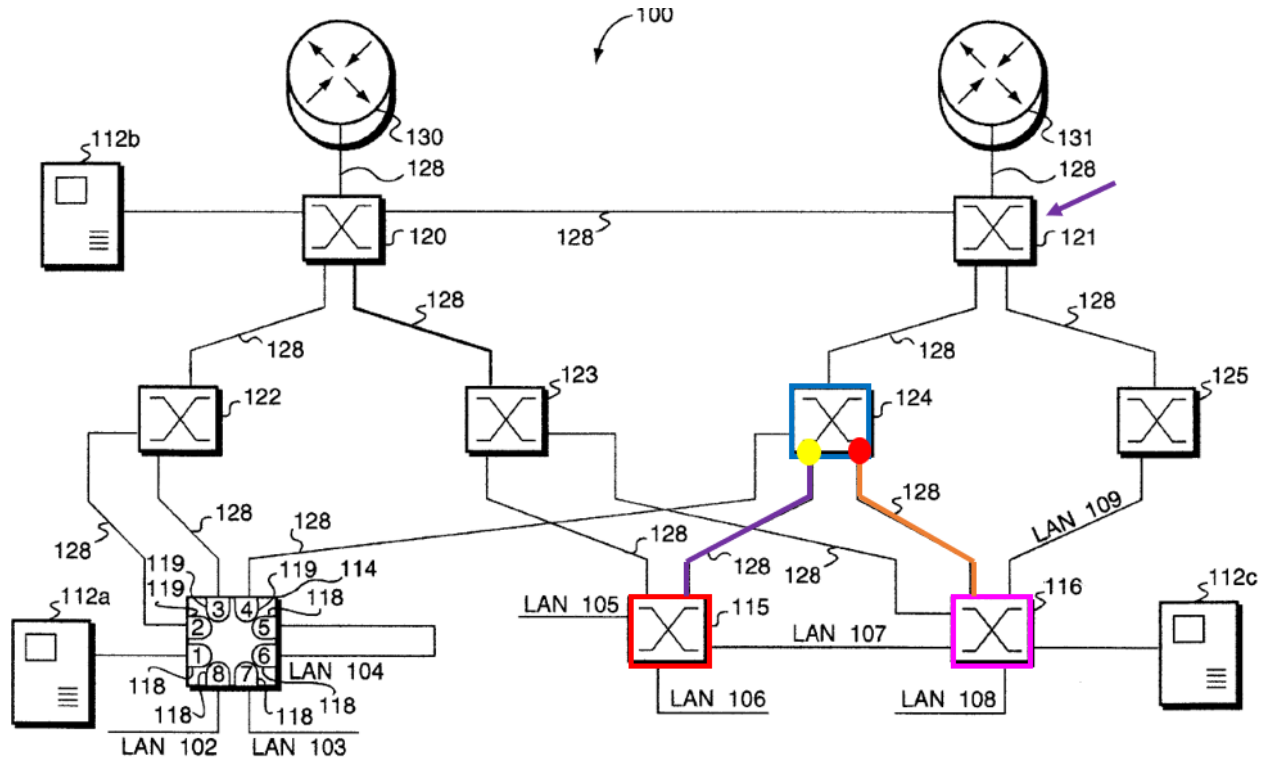


FIG. 1

EX1005, Figure 1.

In fact, EX1006—a continuation application of Gai sharing identical disclosure with Gai—claimed transmitting dummy messages downstream from a newly active designated port, i.e., a port transmitting packets downstream, to inform devices downstream about the new designated port. EX1006, 18:8-11 (“[T]ransmitting, by the back-up designated port, one or more messages, *the messages to tell devices downstream in the spanning tree that the router has a new designated port.*”); see also, EX1006, 17:57-18:11.

Finally, regarding factor (f), Petitioner submits that the additional evidence and facts about the full disclosure of Gai—not only the example discussed in the '986 Patent background section—warrant reconsideration of Gai. Additionally, Dr. Houh is an expert in the relevant art, and his analysis of the prior art has not been presented to the Office. EX1003, ¶¶34-227. In view of the new information presented in this Petition, the Board has ample discretion to institute an IPR on the ground presented.

IX. CONCLUSION

Accordingly, Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Dated: January 30, 2024

Respectfully submitted,

/Theodoros Konstantakopoulos/
Theodoros Konstantakopoulos
(Reg. No. 74,155)
tkonstantakopoulos@desmaraisllp.com
DESMARAIS LLP
230 Park Ave
New York, NY 10169
212-808-1060 (telephone)
212-351-3401 (facsimile)

Counsel for Petitioner

X. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real party-in-interest is Cisco Systems, Inc.

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of the Petitioner, the '968 patent is or was involved in the following case:

Case Heading	Number	Court	Date
<i>Portsmouth Network, Corp. v. Cisco Sys., Inc.</i>	2:23-cv-00441	EDTX	Sept. 26, 2023

C. Lead and Back-up Counsel and Service Information

Lead Counsel

Theodoros Konstantakopoulos
Desmarais LLP
230 Park Ave
New York, NY 10023

Phone: (212) 351-3400
tkonstantakopoulos@desmaraisllp.com
USPTO Reg. No. 74,155

Back-up Counsel

Yung-Hoon Ha
Desmarais LLP
230 Park Ave
New York, NY 10023

Phone: (212) 351-3400
yha@desmaraisllp.com
USPTO Reg. No. 56,368

Please address all correspondence to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above and at
CiscoPortsmouthIPRService@desmaraisllp.com

CERTIFICATION UNDER 37 C.F.R. § 42.24(d)

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and in reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,992. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: January 30, 2024

Respectfully submitted,

/Theodoros Konstantakopoulos/
Theodoros Konstantakopoulos
(Reg. No. 74,155)
tkonstantakopoulos@desmaraisllp.com
DESMARAIS LLP
230 Park Ave
New York, NY 10169
212-808-1060 (telephone)
212-351-3401 (facsimile)

Counsel for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), the undersigned certifies that a complete copy of the foregoing materials:

- Petition for Inter Partes Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104 of U.S.;
- Exhibit List;
- Exhibits 1001-1015;
- Word Count Certification Under 37 CFR § 42.24(d);
- Petitioner's Power of Attorney

was served on counsel of record for Patent Owner by filing this document through P-TACTS and by sending this document via Federal Express Priority Overnight shipping, which is at least as fast and reliable as U.S. Priority Mail Express, to counsel of record for Patent Owner at the correspondence address indicated in Patent Center:

Yehuda Binder
131926 - May Patents Ltd. c/o Dorit Shem-Tov
P.O.B 7230
Ramat-Gan, - 5217102
ISRAEL

Dated: January 30, 2024

Respectfully submitted,

/Theodoros Konstantakopoulos/
Theodoros Konstantakopoulos
(Reg. No. 74,155)
tkonstantakopoulos@desmaraisllp.com
DESMARAIS LLP
230 Park Ave
New York, NY 10169
212-808-1060 (telephone)
212-351-3401 (facsimile)

Counsel for Petitioner