

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

INVALIDITY CLAIM CHART

Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

US Patent No. 7,453,804 (“Feroz”) was issued on November 18, 2008, from an application filed on February 8, 2005. Feroz therefore constitutes prior art under at least 35 U.S.C. §§ 102(a)(1) and 102(a)(2). Feroz anticipates and/or renders obvious the Asserted Claims of the U.S. Patent No. 9,450,884, either alone or in combination with one or more of the references identified in Defendant’s Cover Pleading (at least as Defendant understands WSOU’s application of the Asserted Claims in an effort to show infringement).

To the extent WSOU argues that any element below is not disclosed by Feroz, a person of ordinary skill in the art would have found it obvious to combine the teachings of Feroz with the background knowledge of a person of ordinary skill in the art and/or the additional references, and exemplary teachings, set forth in Appendix E.

The chart below provides representative examples of where each element of each claim is found within Feroz. Citations are meant to be exemplary, not exhaustive, and Defendant reserves the right to identify and discuss additional portions of the reference in support of its contentions and/or to rebut arguments made by WSOU. Citations to figures, drawings, tables, and the like include reference to any accompanying or related text. All internal cross references are meant to incorporate the cross-referenced material as if fully set forth therein.

It is Defendant’s position that WSOU’s Preliminary Infringement Contentions have not established that any accused product or service infringes any valid claim. Thus, Defendant’s statements below should not be treated as an admission, implication, or suggestion that Defendant agrees with WSOU regarding either the scope, construction, or interpretation of any of the Asserted Claims or the infringement theories advanced by WSOU in its Preliminary Infringement Contentions, including whether any Asserted Claim satisfies 35 U.S.C. §§ 101 or 112. In certain cases, Defendant specifies non-limiting examples of where its application of the prior art is based on WSOU’s apparent application of the claim element. These statements are not intended to suggest that Defendant agrees with WSOU’s application of any claim term, suggest a proposed construction at this stage of the case, or suggest that construction is needed, as the parties are not required to exchange terms for construction or proposed constructions until a later date.

WSOU has yet to identify any limitation of the Asserted Claims that it contends is not anticipated and/or rendered obvious by Feroz. Defendant therefore expressly reserves the right to respond to any such contention, including by identifying additional obviousness combinations, if WSOU makes any such contention.

Where Defendant states that Feroz “discloses” a limitation, that disclosure may be express, implicit, and/or inherent.

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
[1pre] A method of adjusting bandwidth allocation by a network switching element in a communications network, the network switching element including a target port, the method comprising:	To the extent the preamble is limiting, Feroz discloses this preamble. <i>See Feroz at Abstract:</i> “Methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates.” <i>See Feroz at 2:45-50:</i> “The present invention relates to management of network resources and, more particularly, to methods, apparatuses and systems directed to an aggregate network resource utilization control mechanism.” <i>See Feroz at 4:46-61:</i> “The present invention provides methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates. These and other aspects of various implementations of the present invention will become apparent from the drawings and description provided below.” <i>See Feroz at 16:56-17:17:</i> “FIG. 6 provides, for didactic purposes, an exemplary hierarchical partition configuration. Root partition node 82 represents the total capacity of access link 21 in either the outbound or inbound direction. In one implementation, separate data structures and processes are maintained for the outbound and inbound directions. Partition nodes 83 represent a first set of partitions allocating the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	capacity of access link at a first level. As FIG. 6 shows, each parent partition may include child partitions represented as partition nodes 83 and 84. For didactic purposes, assume that access link 21 is a T1 line and, thus, provides maximum available bandwidth of 1.5 Mbps. Partitions P1, P2, and P3 may be configured with weights, as discussed above, to equally divide access link 21 into 0.5 Mbps partitions at a first level. In addition, child partitions P4 and P5 may further divide parent partition P1 into a 0.2 Mbps partition (P4) and a 0.3 Mbps partition (P5). Similarly, child partitions P6, P7 and P8 may divide parent partition P2 into two 0.1 Mbps partitions (P6 and P8), and a 0.3 Mbps partition (P7). As discussed above, the initial aggregate bandwidth allocations may also be derived from weights associated with the partition nodes. Of course, any suitable partition configuration may be employed, including the configurations illustrated in FIGS. 9A thru 9F, See above. Still further, other configurations are possible. For example, the root node may represent the access link in both the inbound and outbound direction, while child partitions from the root can correspond to traffic encountered at a given network interface (such as interface 71) and destined for egress from a second network interface (such as interface 72). . . .” <i>See Feroz at 18:28-62:</i> “As FIG. 3 illustrates, packet processor 92 updates, for existing data flows, attributes of the flow object in response to the packet such as the packet count, last packet time, and the like (208). Packet processor 92 can also perform other operations, such as analyzing the packets for connection state information. For example, packet processor 92 can inspect various TCP flags to determine whether the received packet is part of a new data flow or represents a change to an existing data flow (such as the first data packet after the TCP handshake). Methods for determining new data flows and assigning packets to existing data flows are well known in the art and also depend on the particular transport layer protocol employed. For a TCP packet, packet processor 92 can determine a new data flow by detecting SYN and/or SYN/ACK packets. However, a new data flow can simply be a data flow for which there is no corresponding flow object in flow database 135. . . . In one implementation, this separate process can also be passed pointers to TCP FIN packets to adjust various data structures, such as reallocating flow object space to a pool of available memory, adjusting flow counts in traffic class or partition objects.” <i>See Feroz at 18:63-19:26:</i> “As FIG. 3 illustrates, packet processor 92 then passes a pointer to the packet, in one implementation, to traffic classification engine 96, which operates as discussed above to classify the data flow (214). Lastly, the packet is passed to flow control module 94 (218), which performs flow control operations on the data packets. For example, flow control module 94 identifies the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	appropriate partition to which the packet belongs, and applies the partition and per-flow policies to the packet. As FIG. 3 illustrates, application traffic management device 130 may also perform other operations in response to arrival of the packet. For example, measurement engine 140 may record certain measurement variables on an aggregate or per-traffic class basis (218). For example, measurement engine 140 may include one or more traffic class and partition objects that maintain statistics or other parameters used by flow control module 94 to dynamically adjust aggregate and per-flow bandwidth allocations. For example, each partition object, in one implementation, may include the following statistics: 1) a flow count (the number of currently active flows); 2) current rate (a weighted moving average in bps of the rate observed for the flows in the class); 3) aggregate compensation demand; and 4) aggregate unutilized fair share. In addition, per-flow statistics are also maintained, such as the current rate of the flow (a weighted moving average over an analysis interval of the rate at which packets are transmitted from application traffic management device), packet count, and the like. One skilled in the art will recognize that the order of processing operations illustrated in FIG. 3 is one possible process flow and that the flow control processes described herein can be performed at other points in the packet processing path.” <i>See Feroz at 20:6-20:</i> “FIG. 4B sets forth a method, according to one implementation of the invention, directed to computing a per-flow fair share allocation and a target rate for flows within a given partition. As FIG. 4B illustrates, the per-flow fair share bandwidth allocation is the current aggregate bandwidth allocated to the partition divided by the number of active flows (330). As discussed above, the current number of active flows is maintained in a traffic class or partition object. The per-flow fair share rate can be used in a variety of ways. In one implementation, the per-flow fair share rate can be used as the basis of determining whether packets corresponding to a flow are placed on the high or low priority partition queue. In another implementation, the per-flow fair share rate can be used in a process that computes a target rate for each flow within a partition.”
[1a] monitoring, by the network switching element, a data flow traversing the target port of the network switching element;	Feroz discloses monitoring, by the network switching element, a data flow traversing the target port of the network switching element. <i>See Feroz 3:9-54:</i> “To facilitate monitoring, management and control of network environments, a variety of network devices, applications, technologies and services have been developed. For example, certain data flow rate control mechanisms have been developed to provide a means to control and optimize efficiency of data transfer as well as allocate available bandwidth among a variety of business

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	enterprise functionalities. For example, U.S. Pat. No. 6,038,216 discloses a method for explicit data rate control in a packet-based network environment without data rate supervision. Data rate control directly moderates the rate of data transmission from a sending host, resulting in just-in-time data transmission to control inbound traffic and buffering of packets, and reduce the inefficiencies associated with dropped packets. Bandwidth management devices also allow for explicit data rate control for flows associated with a particular traffic classification. For example, U.S. Pat. No. 6,412,000, above, discloses automatic classification of network traffic for use in connection with bandwidth allocation mechanisms. U.S. Pat. No. 6,046,980 discloses systems and methods allowing for application layer control of bandwidth utilization in packet-based computer networks. For example, bandwidth management devices allow network administrators to specify policies operative to control and/or prioritize the bandwidth allocated to individual data flows according to traffic classifications. In addition, certain bandwidth management devices, as well as certain routers, allow network administrators to specify aggregate bandwidth utilization controls to divide available bandwidth into partitions. With some network devices, these partitions can be configured to provide a minimum bandwidth guarantee, and/or cap bandwidth, as to a particular class of traffic. An administrator specifies a traffic class (such as FTP data, or data flows involving a specific user or network application) and the size of the reserved virtual link-i.e., minimum guaranteed bandwidth and/or maximum bandwidth. Such partitions can be applied on a per-application basis (protecting and/or capping bandwidth for all traffic associated with an application) or a per-user basis (controlling, prioritizing, protecting and/or capping bandwidth for a particular user). In addition, certain bandwidth management devices allow administrators to define a partition hierarchy by configuring one or more partitions dividing the access Link and further dividing the parent partitions into one or more child partitions. U.S. patent application Ser. No. 10/108,085 discloses data structures and methods for implementing a partition hierarchy.” See Feroz at 3:55-4:16: “Certain application traffic management devices, such as the PacketShaper® application traffic management device, offered by Packeteer®, Inc. of Cupertino, Calif., support the concurrent use of aggregate bandwidth policies (e.g., partitions), and per-flow bandwidth policies, such as rate policies enforced by the TCP Rate control technologies disclosed in U.S. Pat. No. 6,038,216 [incorporated by reference]. A partition is essentially a bandwidth allocation and queuing mechanism. That is, after a packet processor classifies each packet and pushes each packet onto a partition queue associated with the appropriate partition, another process, typically, loops through the partition queues to pop packets off the queues and populate an output queue. Aggregate

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	bandwidth allocation among the different partitions essentially establishes a preference by which a flow control mechanism arbitrates among the corresponding partition queues. For example, a flow control module, while arbitrating among the partition queues, may read more packets from partitions having a higher allocation of bandwidth relative to partitions that have lower allocations. For example, as disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference above, the bandwidth allocated to a given partition affects the rate at which the partition is selected by an output scheduling process and therefore the length of time packets are buffered in the corresponding partition queue. In addition, TCP Rate Control technologies can be used to effect per-flow rate policies to control or influence the rate at which packets are received at a network device and, therefore, use of inbound network bandwidth and the amount of data that is queued at any given time.” <i>See Feroz at 5:64-6:3:</i> “As discussed more fully below, application traffic management device 130 is operative to classify and manage data flows traversing access link 21. In one implementation, application traffic management device 130 also includes functionality operative to monitor the performance of the network (such as network latency) and/or network applications.” <i>See Feroz at 16:4-25:</i> “As FIG. 2 illustrates, network application traffic management device 130, in one implementation, comprises network device application processor 75, and first and second network interfaces 71, 72, which operably connect application traffic management device 130 to the communications path between router 22 and network 40. Network device application processor 75 generally refers to the functionality implemented by application traffic management device 130, such as network monitoring or reporting, application traffic management, and the like. In one embodiment, network device application processor 75 is a combination of hardware and software, such as a central processing unit, memory, a system bus, an operating system, device drivers, and one or more software modules implementing the functions performed by application traffic management device 130. For didactic purposes, application traffic management device 130 is configured to manage network traffic traversing access link 21. The above-identified patents and patent applications, incorporated by reference herein, disclose various functionalities and features that may be incorporated into application traffic management devices according to various implementations of the present invention.”

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	<i>See Feroz at 6:26-38:</i> “In one embodiment, first and second network interfaces 71, 72 are the hardware communications interfaces that receive and transmit packets over the computer network environment. In one implementation, first and second network interfaces 71, 72 reside on separate network interface cards operably connected to the system bus of application traffic management device 130. In another implementation, first and second network interfaces reside on the same network interface card. In addition, the first and second network interfaces 71, 72 can be wired network interfaces such as Ethernet (IEEE 802.3) interfaces, and/or wireless network interfaces, such as IEEE 802.11, BlueTooth, satellite-based interfaces, and the like.” <i>See Feroz at 6:51-7:9:</i> “As FIG. 2 illustrates, network device application processor 75, in one implementation, includes a packet processor 92, flow control module 94, and traffic classification engine 96. Network device application processor 75, in one implementation, further comprises host database 134, flow database 135, measurement engine 140, management information base 138, and administrator interface 150. In one embodiment, the packet processor 92 is operative to process data packets, such as detecting new data flows, parsing the data packets for various attributes (such by a network administrator, in a hierarchical traffic class structure. In one implementation, flow control module 94 is operative to apply aggregate and per-flow bandwidth utilization controls to data flows traversing the access link 21 in the inbound and/or outbound directions.” <i>See Feroz at 7:10-28:</i> “As discussed above, in one implementation, network device application processor 75 further comprises measurement engine 140, management information base (MIB) 138, and administrator interface 150. Management information base 138 is a database of standard and extended network objects related to the operation of application traffic management device 130. Measurement engine 140 maintains measurement and statistical data relating to operation of application traffic management device 130 to allow for monitoring of bandwidth utilization and network performance across access Link 21 with respect to a plurality of bandwidth utilization and other network statistics on an aggregate and/or per-traffic-class level. In one implementation, measurement engine 140 tracks a variety of metrics corresponding to the partitions implemented by flow control module 94. These metrics allow flow control module 94, as discussed in more detail below, to dynamically adjust bandwidth allocations across flows and partitions to improve or manage network application performance.”

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	<i>See Feroz at 7:43-57:</i> “As discussed above, packet processor 92, in one implementation, is operative to detect new data flows, instantiate data structures associated with the flows and parse packets to identify packet attributes, such as source and destination addresses, port numbers, etc., and populate one or more fields in the data structures. The U.S. patents and patent applications identified above discuss the operation of packet processors that can be incorporated into embodiments of the present invention. In one embodiment, when packet processor 92 encounters a new data flow it stores the source and destination IP addresses contained in the packet headers in host database 134. Packet processor 92 further constructs a control block (flow) object in flow database 135 including attributes characterizing a specific flow between two end systems, such as source and destination port numbers, etc.” <i>See Feroz at 8:6-14:</i> “As discussed more fully below, in one embodiment, flows are identified based on the following flow attributes: 1) source IP address, 2) destination IP address, 3) source port number, 4) destination port number, and 5) protocol (derived from the “protocol” field in IPv4 headers, and the “NextHeader” field in IPv6 headers). One skilled in the art will recognize that flows can be identified in relation to a variety of attributes and combinations of attributes.” <i>See Feroz at 8:14-42:</i> “In addition, methods for determining new data flows and assigning packets to existing data flows are well known in the art and also depend on the particular transport layer protocol employed. For a TCP flow, for example, packet processor 92 can determine a new data flow by detecting SYN, SYN/ACK, and/or ACK packets. . . . The termination of TCP connections is typically detected by identifying FIN packets; however, the timeout mechanisms discussed above can be used in situations where a FIN packet is not detected.” <i>See Feroz at 8:61-9:9:</i> “For a TCP/IP packet, packet processor 92 can compute the inside and outside addresses based on the source and destination network addresses of the packet and the direction of the packet flow. Still further, packet processor 92 can also identify which host is the client and which host is the server for a given data flow and store this information in the flow specification or flow object. The identification of a server or client in a given transaction generally depends on the network protocols employed by the hosts. For example, in TCP flows, a client initiates a transaction by transmitting a SYN packet to initiate a TCP connection. Application traffic management device 130 can detect the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	SYN packet and note the source network address of the packet as the client host, and the destination address as the server host. One of ordinary skill in the art will recognize how to identify clients and servers in connection with other networking protocols.” <i>See Feroz at 9:10-36:</i> “In one embodiment, packet processor 92 creates and stores flow objects corresponding to data flows in flow database 135. In one embodiment, flow object attributes include a pointer to a corresponding flow specification object, as well as other flow state parameters, such as TCP connection status, timing of last packets in the inbound and outbound directions, speed information, apparent round trip time, packet count, etc. Flow object attributes further include at least one traffic class identifier (or pointer(s) thereto) associated with the data flow, as well as policy parameters (or pointers thereto) corresponding to the identified traffic class. In one embodiment, flow objects further include a list of traffic classes for which measurement data (maintained by measurement engine 140) associated with the data flow should be logged. In one embodiment, to facilitate association of an existing flow object to subsequent packets associated with a data flow or connection, flow database 135 further maintains a control block hash table including a key comprising a hashed value computed from a string comprising the inside IP address, outside IP address, inside port number, outside port number, and protocol type (e.g., TCP, UDP, etc.) associated with a pointer to the corresponding flow object. According to this embodiment, to identify whether a flow object exists for a given data flow, packet processor 92 hashes the values identified above and scans the hash table for a matching entry. If one exists, packet processor 92 associates the pointer to the corresponding flow object with the packets in the data flow.” <i>See Feroz at 9:38-45:</i> “As discussed above, traffic classification engine 96, in one implementation, is operative to classify data flows into one of a plurality of traffic classes. Traffic classification engine 96, in one implementation, comprises a plurality of service type identification modules, each of which correspond to a set of service types. Each service type identification module analyzes one or more packets in a given data flow to attempt to identify a service type corresponding to the flow.” <i>See Feroz at Feroz at 11:13-23:</i> “A traffic class comprises a set of matching rules or attributes allowing for logical grouping of data flows that share the same characteristic or set of characteristics. In one implementation, the matching rules can correspond to the service type identifiers discussed above, as well as other data flow attributes, such as the network interface on which the packets are received by application

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	traffic management device 130, whether the server is the inside or outside host (see above), non-standard and standard port numbers, host IP address or subnet, MAC address, application-specific strings, diffserv codes, MPLS tags, VLAN tags, and the like.” <i>See Feroz at 12:8-13:</i> “Traffic classification engine 96, in one implementation, stores traffic classes associated with data flows that traverse access link 21. Traffic classification engine 96, in one embodiment, stores the traffic classes and corresponding data (e.g., matching rules, policies, partition pointers, etc.) related to each traffic class in a hierarchical tree.” <i>See Feroz at 13:7-38:</i> “As discussed more fully below, flow control module 94 enforces aggregate, partition-based bandwidth utilization controls on data flows traversing access link 21. In one implementation, flow control module 94 monitors bandwidth demand of each partition relative to an aggregate, weighted fair share allocation scheme, and dynamically re-distributes unutilized bandwidth to partitions whose demand for bandwidth exceeds its weighted fair share allocation. As discussed in more detail below, flow control module 94 performs these dynamic allocation adjustments on a continuous basis, as needed. In one implementation, the demand for bandwidth at a given partition can be evaluated in a number of ways. For example, the size of, and rate at which, packets in a flow are received by application traffic management device 130 during a given time interval can be monitored to compute a per-flow rate demand. The per-flow rate demands for all flows falling within a given partition can be aggregated to yield an aggregate rate demand for the partition. In addition, these computed rate values can be smoothed by maintaining a rolling average or weighted moving average across a sliding window of time. Aggregate rate demand for a partition may also be determined by analyzing the number of packets queued in a corresponding partition queue over a given analysis interval. For example, the change in the number of packets queued in a given partition queue over an analysis interval can be analyzed to determine the aggregate rate demand for a given partition. As discussed in more detail below, flow control module 94 can then dynamically allocate unutilized bandwidth to partitions where the aggregate rate demand exceeds their respective bandwidth allocations. In one implementation, the weighting values associated with the partitions determines in part how unutilized bandwidth is distributed.” <i>See Feroz at 13:39-64:</i> “In addition, flow control module 94, in some implementations, can optionally apply per-flow bandwidth utilization controls, on data flows traversing access link 21. In one implementation, flow

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	control module 94 includes a partitioning module operative to enforce aggregate bandwidth utilization controls (e.g., partitions), and a per-flow rate control module operative to apply per-flow rate controls on data flows. In one implementation, flow control module 94 implements the deterministic partition scheduling functionality disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference herein. Furthermore, as discussed more fully below, flow control module 94 includes functionality for dynamically adjusting aggregate bandwidth allocation among partitions. In one implementation, flow control module 94 also includes processes that dynamically adjust per-flow bandwidth allocation within partitions. A partition operates to manage bandwidth for aggregate data flows associated with a traffic class. As discussed above, a partition is essentially a division of the capacity of access link 21. Still further, flow control module 94, in some implementations, may also implement per-flow rate policies on the data flows. In one implementation, flow control module 94 implements the TCP Rate Control technologies disclosed in U.S. Pat. No. 6,038,216 to control the rate at which transmitters send data and therefore the amount of data that is queued in buffers at application traffic management device 130.” <i>See Feroz at 17:64-18:7:</i> Incorporated by reference U.S. Pat. No. 6,038,216 discloses “the window size is set to allow the specified rate given the current round trip time (“RTT”) and the detected speed.” “FIG. 3 illustrates the overall process flow, according to one implementation of the present invention, directed to the operation of application traffic management devices 130. In one embodiment, packet processor 92 receives a data packet (FIG. 3, 202) and determines whether flow database 135 contains an existing flow object corresponding to the data flow (204) (see Section A.1., supra). If no flow object corresponds to the data packet, packet processor 92 constructs a flow object including attributes characterizing the data flow, such as source address, destination address, etc. (206) (see above).” <i>See Feroz at 18:28-62:</i> “As FIG. 3 illustrates, packet processor 92 updates, for existing data flows, attributes of the flow object in response to the packet such as the packet count, last packet time, and the like (208). Packet processor 92 can also perform other operations, such as analyzing the packets for connection state information. For example, packet processor 92 can inspect various TCP flags to determine whether the received packet is part of anew data flow or represents a change to an existing data flow (such as the first data packet after the TCP handshake). Methods for determining new data flows and assigning packets to existing data flows are well known in the art and also depend on the particular transport layer protocol employed. For a TCP packet, packet processor 92 can determine a new data

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	flow by detecting SYN and/or SYN/ACK packets. However, a new data flow can simply be a data flow for which there is no corresponding flow object in flow database 135.... In one implementation, this separate process can also be passed pointers to TCP FIN packets to adjust various data structures, such as reallocating flow object space to a pool of available memory, adjusting flow counts in traffic class or partition objects.” <i>See Feroz at 19:12-26:</i> “For example, each partition object, in one implementation, may include the following statistics: 1) a flow count (the number of currently active flows); 2) current rate (a weighted moving average in bps of the rate observed for the flows in the class); 3) aggregate compensation demand; and 4) aggregate unutilized fair share. In addition, per-flow statistics are also maintained, such as the current rate of the flow (a weighted moving average over an analysis interval of the rate at which packets are transmitted from application traffic management device), packet count, and the like. One skilled in the art will recognize that the order of processing operations illustrated in FIG. 3 is one possible process flow and that the flow control processes described herein can be performed at other points in the packet processing path.”
[1b] determining, by the network switching element, a bandwidth allocation for the target port, the bandwidth allocation for the target port being a bandwidth that is currently allocated for the data flow;	Feroz discloses determining, by the network switching element, a bandwidth allocation for the target port, the bandwidth allocation for the target port being a bandwidth that is currently allocated for the data flow. <i>See Feroz at Abstract:</i> “Methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates.” <i>See Feroz 3:9-54:</i> “To facilitate monitoring, management and control of network environments, a variety of network devices, applications, technologies and services have been developed. For example, certain data flow rate control mechanisms have been developed to provide a means to control and optimize

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	efficiency of data transfer as well as allocate available bandwidth among a variety of business enterprise functionalities. For example, U.S. Pat. No. 6,038,216 discloses a method for explicit data rate control in a packet-based network environment without data rate supervision. Data rate control directly moderates the rate of data transmission from a sending host, resulting in just-in-time data transmission to control inbound traffic and buffering of packets, and reduce the inefficiencies associated with dropped packets. Bandwidth management devices also allow for explicit data rate control for flows associated with a particular traffic classification. For example, U.S. Pat. No. 6,412,000, above, discloses automatic classification of network traffic for use in connection with bandwidth allocation mechanisms. U.S. Pat. No. 6,046,980 discloses systems and methods allowing for application layer control of bandwidth utilization in packet-based computer networks. For example, bandwidth management devices allow network administrators to specify policies operative to control and/or prioritize the bandwidth allocated to individual data flows according to traffic classifications. In addition, certain bandwidth management devices, as well as certain routers, allow network administrators to specify aggregate bandwidth utilization controls to divide available bandwidth into partitions. With some network devices, these partitions can be configured to provide a minimum bandwidth guarantee, and/or cap bandwidth, as to a particular class of traffic. An administrator specifies a traffic class (such as FTP data, or data flows involving a specific user or network application) and the size of the reserved virtual link-i.e., minimum guaranteed bandwidth and/or maximum bandwidth. Such partitions can be applied on a per-application basis (protecting and/or capping bandwidth for all traffic associated with an application) or a per-user basis (controlling, prioritizing, protecting and/or capping bandwidth for a particular user). In addition, certain bandwidth management devices allow administrators to define a partition hierarchy by configuring one or more partitions dividing the access Link and further dividing the parent partitions into one or more child partitions. U.S. patent application Ser. No. 10/108,085 discloses data structures and methods for implementing a partition hierarchy.” See Feroz at 3:55-4:16: “Certain application traffic management devices, such as the PacketShaper® application traffic management device, offered by Packeteer®, Inc. of Cupertino, Calif., support the concurrent use of aggregate bandwidth policies (e.g., partitions), and per-flow bandwidth policies, such as rate policies enforced by the TCP Rate control technologies disclosed in U.S. Pat. No. 6,038,216 [incorporated by reference]. A partition is essentially a bandwidth allocation and queuing mechanism. That is, after a packet processor classifies each packet and pushes each packet onto a partition queue associated with the appropriate partition, another process, typically, loops through

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	the partition queues to pop packets off the queues and populate an output queue. Aggregate bandwidth allocation among the different partitions essentially establishes a preference by which a flow control mechanism arbitrates among the corresponding partition queues. For example, a flow control module, while arbitrating among the partition queues, may read more packets from partitions having a higher allocation of bandwidth relative to partitions that have lower allocations. For example, as disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference above, the bandwidth allocated to a given partition affects the rate at which the partition is selected by an output scheduling process and therefore the length of time packets are buffered in the corresponding partition queue. In addition, TCP Rate Control technologies can be used to effect per-flow rate policies to control or influence the rate at which packets are received at a network device and, therefore, use of inbound network bandwidth and the amount of data that is queued at any given time.” <i>See Feroz at 4:17-36:</i> “While partitions and per-flow rate policies are effective for their intended purposes, the configuration parameters that lead to their effective and predictable concurrent operation is often beyond the skill level of the average end user. For example, effective configuration of per-flow rate policies within a partition often requires a general knowledge of the typical number of data flows falling within the partition at given times. For example, a per-flow target rate is generally allocated based on an estimated rate demand for a given flow, regardless of the configuration of the partition that controls the flows in the aggregate. This per-flow target rate may be limited or adjusted by a per-flow rate policy configured by a user. In either case, these circumstances can result in significant under utilization of available bandwidth if the data flows do not actually consume their allocated rate, resulting in under utilization of the bandwidth allocated to the partition and the access link itself. Furthermore, use of explicit per-flow rate policies as the primary means of allocating per-flow bandwidth may not achieve a fair distribution of this bandwidth across the active flows in the partition.” <i>See Feroz at 4:46-61:</i> “The present invention provides methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates. These and other aspects of various implementations of the present invention will become apparent from the drawings and description provided below.” <i>See Feroz at 5:49-6:3:</i> “... As FIGS. 1 and 2 show, the first network 40 interconnects several TCP/IP end systems, including client devices 42 and server device 44, and provides access to resources operably connected to computer network 50 via router 22 and access link 21. Access link 21 is a physical and/or logical connection between two networks, such as computer network 50 and network 40. The computer network environment, including network 40 and network 50 is a packet-based communications environment, employing TCP/IP protocols, and/or other suitable protocols, and has a plurality of interconnected digital packet transmission stations or routing nodes. First network 40, and networks 40 a & 40 b, can each be a local area network, a wide area network, or any other suitable network. As FIGS. 1 and 2 illustrate, application traffic management device 130, in one implementation, is deployed at the edge of network 40. As discussed more fully below, application traffic management device 130 is operative to classify and manage data flows traversing access link 21. In one implementation, application traffic management device 130 also includes functionality operative to monitor the performance of the network (such as network latency) and/or network applications.” <i>See Feroz at 6:4-25:</i> “As FIG. 2 illustrates, network application traffic management device 130, in one implementation, comprises network device application processor 75, and first and second network interfaces 71, 72, which operably connect application traffic management device 130 to the communications path between router 22 and network 40. Network device application processor 75 generally refers to the functionality implemented by application traffic management device 130, such as network monitoring or reporting, application traffic management, and the like. In one embodiment, network device application processor 75 is a combination of hardware and software, such as a central processing unit, memory, a system bus, an operating system, device drivers, and one or more software modules implementing the functions performed by application traffic management device 130. For didactic purposes, application traffic management device 130 is configured to manage network traffic traversing access link 21. The above-identified patents and patent applications, incorporated by reference herein, disclose various functionalities and features that may be

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	incorporated into application traffic management devices according to various implementations of the present invention.” <i>See Feroz at 6:26-50:</i> “In one embodiment, first and second network interfaces 71, 72 are the hardware communications interfaces that receive and transmit packets over the computer network environment. In one implementation, first and second network interfaces 71, 72 reside on separate network interface cards operably connected to the system bus of application traffic management device 130. In another implementation, first and second network interfaces reside on the same network interface card. In addition, the first and second network interfaces 71, 72 can be wired network interfaces, such as Ethernet (IEEE 802.3) interfaces, and/or wireless network interfaces, such as IEEE 802.11, BlueTooth, satellite-based interfaces, and the like. As FIG. 2 illustrates, application traffic management device 130, in one embodiment, includes persistent memory 76, such as a hard disk drive or other suitable memory device, such as writable CD, DVD, or tape drives. In other implementations, application traffic management device 130 can include additional network interfaces, beyond network interfaces 71 and 72, to support additional access links or other functionality. Furthermore, U.S. application Ser. No. 10/843,185 provides a description of the operation of various modules (according to one possible implementation of the present invention), such as network interface drivers, and data structures for receiving into memory and processing packets encountered at network interfaces 71, 72.” <i>See Feroz at 6:51-7:9:</i> “As FIG. 2 illustrates, network device application processor 75, in one implementation, includes a packet processor 92, flow control module 94, and traffic classification engine 96. Network device application processor 75, in one implementation, further comprises host database 134, flow database 135, measurement engine 140, management information base 138, and administrator interface 150. In one embodiment, the packet processor 92 is operative to process data packets, such as detecting new data flows, parsing the data packets for various attributes (such by a network administrator, in a hierarchical traffic class structure. In one implementation, flow control module 94 is operative to apply aggregate and per-flow bandwidth utilization controls to data flows traversing the access link 21 in the inbound and/or outbound directions.” <i>See Feroz at 7:10-28:</i> “As discussed above, in one implementation, network device application processor 75 further comprises measurement engine 140, management information base (MIB) 138, and administrator

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	interface 150. Management information base 138 is a database of standard and extended network objects related to the operation of application traffic management device 130. Measurement engine 140 maintains measurement and statistical data relating to operation of application traffic management device 130 to allow for monitoring of bandwidth utilization and network performance across access Link 21 with respect to a plurality of bandwidth utilization and other network statistics on an aggregate and/or per-traffic-class level. In one implementation, measurement engine 140 tracks a variety of metrics corresponding to the partitions implemented by flow control module 94. These metrics allow flow control module 94, as discussed in more detail below, to dynamically adjust bandwidth allocations across flows and partitions to improve or manage network application performance.” See Feroz at 7:43-48: “As discussed above, packet processor 92, in one implementation, is operative to detect new data flows, instantiate data structures associated with the flows and parse packets to identify packet attributes, such as source and destination addresses, port numbers, etc., and populate one or more fields in the data structures. . . .” See Feroz at 8:6-12: “As discussed more fully below, in one embodiment, flows are identified based on the following flow attributes: 1) source IP address, 2) destination IP address, 3) source port number, 4) destination port number, and 5) protocol (derived from the “protocol” field in IPv4 headers, and the “NextHeader” field in IPv6 headers).” See Feroz at 8:17-42: “For a TCP flow, for example, packet processor 92 can determine a new data flow by detecting SYN, SYN/ACK, and/or ACK packets. . . . The termination of TCP connections is typically detected by identifying FIN packets; however, the timeout mechanisms discussed above can be used in situations where a FIN packet is not detected.” See Feroz at 8:61-9:9: “For a TCP/IP packet, packet processor 92 can compute the inside and outside addresses based on the source and destination network addresses of the packet and the direction of the packet flow. Still further, packet processor 92 can also identify which host is the client and which host is the server for a given data flow and store this information in the flow specification or flow object. The identification of a server or client in a given transaction generally depends on the network protocols

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	employed by the hosts. For example, in TCP flows, a client initiates a transaction by transmitting a SYN packet to initiate a TCP connection. Application traffic management device 130 can detect the SYN packet and note the source network address of the packet as the client host, and the destination address as the server host. One of ordinary skill in the art will recognize how to identify clients and servers in connection with other networking protocols.” <i>See Feroz at 9:10-36:</i> “In one embodiment, packet processor 92 creates and stores flow objects corresponding to data flows in flow database 135. In one embodiment, flow object attributes include a pointer to a corresponding flow specification object, as well as other flow state parameters, such as TCP connection status, timing of last packets in the inbound and outbound directions, speed information, apparent round trip time, packet count, etc. Flow object attributes further include at least one traffic class identifier (or pointer(s) thereto) associated with the data flow, as well as policy parameters (or pointers thereto) corresponding to the identified traffic class. In one embodiment, flow objects further include a list of traffic classes for which measurement data (maintained by measurement engine 140) associated with the data flow should be logged. In one embodiment, to facilitate association of an existing flow object to subsequent packets associated with a data flow or connection, flow database 135 further maintains a control block hash table including a key comprising a hashed value computed from a string comprising the inside IP address, outside IP address, inside port number, outside port number, and protocol type (e.g., TCP, UDP, etc.) associated with a pointer to the corresponding flow object. According to this embodiment, to identify whether a flow object exists for a given data flow, packet processor 92 hashes the values identified above and scans the hash table for a matching entry. If one exists, packet processor 92 associates the pointer to the corresponding flow object with the packets in the data flow.” <i>See Feroz at 9:38-45:</i> “As discussed above, traffic classification engine 96, in one implementation, is operative to classify data flows into one of a plurality of traffic classes. Traffic classification engine 96, in one implementation, comprises a plurality of service type identification modules, each of which correspond to a set of service types. Each service type identification module analyzes one or more packets in a given data flow to attempt to identify a service type corresponding to the flow.” <i>See Feroz at Feroz at 11:13-23:</i> “A traffic class comprises a set of matching rules or attributes allowing for logical grouping of data flows that share the same characteristic or set of characteristics. In one implementation, the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	matching rules can correspond to the service type identifiers discussed above, as well as other data flow attributes, such as the network interface on which the packets are received by application traffic management device 130, whether the server is the inside or outside host (see above), non-standard and standard port numbers, host IP address or subnet, MAC address, application-specific strings, diffserv codes, MPLS tags, VLAN tags, and the like.” <i>See Feroz at 12:8-13:</i> “Traffic classification engine 96, in one implementation, stores traffic classes associated with data flows that traverse access link 21. Traffic classification engine 96, in one embodiment, stores the traffic classes and corresponding data (e.g., matching rules, policies, partition pointers, etc.) related to each traffic class in a hierarchical tree.” <i>See Feroz at 13:7-38:</i> “As discussed more fully below, flow control module 94 enforces aggregate, partition-based bandwidth utilization controls on data flows traversing access link 21. In one implementation, flow control module 94 monitors bandwidth demand of each partition relative to an aggregate, weighted fair share allocation scheme, and dynamically re-distributes unutilized bandwidth to partitions whose demand for bandwidth exceeds its weighted fair share allocation. As discussed in more detail below, flow control module 94 performs these dynamic allocation adjustments on a continuous basis, as needed. In one implementation, the demand for bandwidth at a given partition can be evaluated in a number of ways. For example, the size of, and rate at which, packets in a flow are received by application traffic management device 130 during a given time interval can be monitored to compute a per-flow rate demand. The per-flow rate demands for all flows falling within a given partition can be aggregated to yield an aggregate rate demand for the partition. In addition, these computed rate values can be smoothed by maintaining a rolling average or weighted moving average across a sliding window of time. Aggregate rate demand for a partition may also be determined by analyzing the number of packets queued in a corresponding partition queue over a given analysis interval. For example, the change in the number of packets queued in a given partition queue over an analysis interval can be analyzed to determine the aggregate rate demand for a given partition. As discussed in more detail below, flow control module 94 can then dynamically allocate unutilized bandwidth to partitions where the aggregate rate demand exceeds their respective bandwidth allocations. In one implementation, the weighting values associated with the partitions determines in part how unutilized bandwidth is distributed.”

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	<i>See Feroz at 13:39-64:</i> “In addition, flow control module 94, in some implementations, can optionally apply per-flow bandwidth utilization controls, on data flows traversing access link 21. In one implementation, flow control module 94 includes a partitioning module operative to enforce aggregate bandwidth utilization controls (e.g., partitions), and a per-flow rate control module operative to apply per-flow rate controls on data flows. In one implementation, flow control module 94 implements the deterministic partition scheduling functionality disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference herein. Furthermore, as discussed more fully below, flow control module 94 includes functionality for dynamically adjusting aggregate bandwidth allocation among partitions. In one implementation, flow control module 94 also includes processes that dynamically adjust per-flow bandwidth allocation within partitions. A partition operates to manage bandwidth for aggregate data flows associated with a traffic class. As discussed above, a partition is essentially a division of the capacity of access link 21. Still further, flow control module 94, in some implementations, may also implement per-flow rate policies on the data flows. In one implementation, flow control module 94 implements the TCP Rate Control technologies disclosed in U.S. Pat. No. 6,038,216 to control the rate at which transmitters send data and therefore the amount of data that is queued in buffers at application traffic management device 130.” <i>See Feroz at 14:13-20:</i> Incorporated by reference U.S. Pat. No. 6,038,216 discloses “the window size is set to allow the specified rate given the current round trip time ("RTT") and the detected speed.” “Similar to the traffic class tree, the root partition node represents the capacity of the access link in either the inbound or outbound directions. In one implementation, separate partition configurations and controls can be maintained for data flows traveling in the inbound or outbound directions relative to network 40. In other implementations, a single partition configuration can be applied to both inbound and outbound data flows.” <i>See Feroz at 14:21-41:</i> “Once a partition configuration is established, in one implementation, a user can simply assign a weight to each partition node, which essentially expresses the priority of the network traffic class(es) falling within the partition relative to other traffic classes. Optionally, a user may also assign a minimum aggregate bandwidth allocation to one or more of the selected partition nodes. As discussed in more detail below, each partition is then allocated a fair share of available bandwidth based on the weights assigned to the partitions, limited by the minimum and maximum bandwidth settings configured for any partitions. In another implementation, a user may also

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	specify a maximum bandwidth allocation for one or more partitions. In one implementation, unless the user specifically configures a weighting value for at least one partition in a given level of the hierarchy, by default, each sibling partition is weighted equally. Additionally, if a user configures a weighting value for only one partition, each sibling partition is configured with a default weighting value. In one implementation, administrator interface 150 requires the user to enter weighting values for all sibling partitions after a user initially configures a weighting value for a selected partition.” See Feroz at 14:42-58: “After the user has completed a configuration, administrator interface 150, in one implementation, processes the configuration to create an aggregate bandwidth allocation scheme that includes partitions, and optionally per-flow rate policies, based on the received configuration. For example, partition queues are created for each leaf partition, and weighted, fair share bandwidth allocations are computed for the partitions, as discussed in more detail below. In addition, in one implementation, each active data flow within a given partition is also guaranteed a fair share of the bandwidth allocated to the partition. The aggregate fair share allocation for each partition is generally stable, unless there are configuration changes. However, the fair share allocation for each flow changes dynamically based on the number of active flows in the partition. In addition, the actual bandwidth allocated to a given partition at a given point in time may change dynamically in response to network traffic demands” See Feroz at 15:6-14: $\text{Aggregate Weight} = \sum \text{Weights from Child Nodes}$ $\text{Allocation Per Partition } (c) = \frac{\text{Parent Allocation} * \text{Configured Weight of Partition } (c)}{\text{Aggregate Weight}}$

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	<i>See Feroz at 16:56-17:17:</i> “FIG. 6 provides, for didactic purposes, an exemplary hierarchical partition configuration. Root partition node 82 represents the total capacity of access link 21 in either the outbound or inbound direction. In one implementation, separate data structures and processes are maintained for the outbound and inbound directions. Partition nodes 83 represent a first set of partitions allocating the capacity of access link at a first level. As FIG. 6 shows, each parent partition may include child partitions represented as partition nodes 83 and 84. For didactic purposes, assume that access link 21 is a T1 line and, thus, provides maximum available bandwidth of 1.5 Mbps. Partitions P1, P2, and P3 may be configured with weights, as discussed above, to equally divide access link 21 into 0.5 Mbps partitions at a first level. In addition, child partitions P4 and P5 may further divide parent partition P1 into a 0.2 Mbps partition (P4) and a 0.3 Mbps partition (P5). Similarly, child partitions P6, P7 and P8 may divide parent partition P2 into two 0.1 Mbps partitions (P6 and P8), and a 0.3 Mbps partition (P7). As discussed above, the initial aggregate bandwidth allocations may also be derived from weights associated with the partition nodes. Of course, any suitable partition configuration may be employed, including the configurations illustrated in FIGS. 9A thru 9F, See above. Still further, other configurations are possible. For example, the root node may represent the access link in both the inbound and outbound direction, while child partitions from the root can correspond to traffic encountered at a given network interface (such as interface 71) and destined for egress from a second network interface (such as interface 72). . . .” <i>See Feroz at 18:18-27:</i> “In addition, packet processor 92 also computes an estimated initial data rate for the flow (207). In one implementation, packet processor 92 uses the rapid data rate detection technologies disclosed in U.S. Pat. No. 5,802,106 to provide an initial estimate of the data rate for the flow. In one implementation, the estimated data rate is stored in a field of the flow object. In one implementation, this initial rate estimate is replaced, as more packets in the flow traverse application traffic management device 130, by a current rate based on weighted moving average over an analysis interval.” <i>See Feroz at 18:28-62:</i> “FIG. 3 illustrates, packet processor 92 updates, for existing data flows, attributes of the flow object in response to the packet such as the packet count, last packet time, and the like (208). Packet processor 92 can also perform other operations, such as analyzing the packets for connection state information. For example, packet processor 92 can inspect various TCP flags to determine whether the received packet is part of a new data flow or represents a change to an existing data flow (such

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	as the first data packet after the TCP handshake). Methods for determining new data flows and assigning packets to existing data flows are well known in the art and also depend on the particular transport layer protocol employed. For a TCP packet, packet processor 92 can determine a new data flow by detecting SYN and/or SYN/ACK packets. However, a new data flow can simply be a data flow for which there is no corresponding flow object in flow database 135. . . . In one implementation, this separate process can also be passed pointers to TCP FIN packets to adjust various data structures, such as reallocating flow object space to a pool of available memory, adjusting flow counts in traffic class or partition objects.” <i>See Feroz at 19:12-26:</i> “For example, each partition object, in one implementation, may include the following statistics: 1) a flow count (the number of currently active flows); 2) current rate (a weighted moving average in bps of the rate observed for the flows in the class); 3) aggregate compensation demand; and 4) aggregate unutilized fair share. In addition, per-flow statistics are also maintained, such as the current rate of the flow (a weighted moving average over an analysis interval of the rate at which packets are transmitted from application traffic management device), packet count, and the like. One skilled in the art will recognize that the order of processing operations illustrated in FIG. 3 is one possible process flow and that the flow control processes described herein can be performed at other points in the packet processing path.” <i>See Feroz at 19:28-20:4:</i> “FIG. 4A illustrates a process, according to one implementation of the invention, directed to reading packets onto flow and partition queues. . . . In addition, since the packet represents the addition of a new flow to a partition, flow control module 94 re-computes the per-flow bandwidth allocation for the partition (316), as discussed in a following section. . . . In one implementation, if the initial flow rate of the flow is less than the per-flow fair share for the partition (318), flow control module 94 pushes the flow queue onto the high priority queue (322), facilitating the chance for the rate of the flow to increase. Otherwise, flow control module 94 pushes the flow queue onto the low priority queue 60 a corresponding to the identified partition (320). In one implementation, flow control module 94 applies a process directed to the per-flow rate enforcement. In one implementation, the per-flow target rate is enforced by mapping the target rate to an advertised TCP window size, as disclosed in U.S. Pat. No. 6,038,216, and modifying the advertised window size in the TCP ACK fields of packets transmitted back to the transmitter to modify the rate at which subsequent packets are sent. Of course, the per-flow rate control processes may be applied to the packets at any point prior to egress from application traffic management device 130. In addition, the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	use of TCP Rate Control technologies to control the rate of data transmission from end hosts is an optional feature.” <i>See Feroz at 20:6-20:</i> “FIG. 4B sets forth a method, according to one implementation of the invention, directed to computing a per-flow fair share allocation and a target rate for flows within a given partition. As FIG. 4B illustrates, the per-flow fair share bandwidth allocation is the current aggregate bandwidth allocated to the partition divided by the number of active flows (330). As discussed above, the current number of active flows is maintained in a traffic class or partition object. The per-flow fair share rate can be used in a variety of ways. In one implementation, the per-flow fair share rate can be used as the basis of determining whether packets corresponding to a flow are placed on the high or low priority partition queue. In another implementation, the per-flow fair share rate can be used in a process that computes a target rate for each flow within a partition.” <i>See Feroz at 20:21-45</i> “According to one implementation, the allocation of bandwidth to flows within a given partition is an optional step. In such an implementation, as FIG. 4B illustrates, flow control module 94 computes the aggregate compensation demand and the aggregate bandwidth underutilization for the partition (331). Compensation demand occurs when the current rate of a flow exceeds a threshold based on a percentage of the target rate of the flow. In one implementation, the compensation demand threshold is 94% of the per-flow target rate; however, this percentage can vary based on various engineering and design considerations. The compensation demand for a flow, in one implementation, is a fixed percentage of the detected current rate for the flow. As discussed above, the current rate can be estimated, in one implementation, by monitoring the size of incoming packets and the time at which they are received. Aggregate compensation demand for a partition is the sum of the compensation demands for each flow. Oppositely, bandwidth underutilization (unutilized fair share) occurs when the current rate associated with the flow is less than a percentage of the target rate, and is computed based on the difference between the per-flow fair share bandwidth allocation and the current rate. In one implementation, the aggregate bandwidth underutilization for a partition is thus the sum of the individual bandwidth underutilization values for the flows in the partition.” <i>See Feroz at 20:46-21:15:</i> “As FIG. 4B illustrates, flow control module 94, in one implementation, distributes some of the unutilized bandwidth within a partition to flows having compensation demand in the partition. In

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	one implementation, flow control module 94, for all active flows in the partition (332), checks whether the current rate of the flow is less than X percent of the target rate of the flow (336). As discussed above, in one implementation, X=94. As FIG. 4B illustrates, the target rate for new flows (333) is the per-flow fair share previously computed (330). If the current rate of the flow is less than X percent of the target rate, flow control module 94 then determines whether a flow compensation demand has been previously allocated to the flow in a previous cycle (344). If so, flow control module 94 then determines whether the current rate is less than Y percent of the target rate (346). In one implementation, Y equals 88; however, other values can be used depending on engineering and design considerations. If both 344 and 346 are met, flow control module 94 steps the flow compensation demand (FCD) down by Z percent of the previously allocated flow compensation demand (348). In one implementation, Z equals 6 percent; however, other values can be used. While other implementations are contemplated by the invention, the foregoing prevents temporary fluctuations in data rates from substantially affecting the flow target rate, thereby allowing the rate of the flow to potentially increase. Otherwise, if a flow compensation demand was not previously allocated to the flow (344), flow control module 94 assigns a target rate equal to the lesser of 1) the current rate of the flow multiplied by a factor M, which in one implementation is 1.5; or 2) the per-flow fair share bandwidth allocation (349). As FIG. 4B illustrates, the excess bandwidth (Fair Share less Current Rate), if any, is passed up to the partition to enable subsequent calculations of aggregate bandwidth underutilization. Assigning a target rate that is greater than the measured current rate helps to avoid the recursive spiral by which the flow data rate decreases, allowing the flow data rate to increase up to the fair share allocation.” See Feroz at 22:4-24: “In one implementation, flow control module 94 includes an aggregate bandwidth allocation daemon that operates at periodic intervals (e.g., 125 milliseconds, etc.) to update the aggregate bandwidth allocation across the partitions based on the weighted fair share allocations, and the observed aggregate compensation demand and aggregate bandwidth underutilization. In one implementation, flow control module 94 distributes unutilized or excess bandwidth to partitions whose demand for bandwidth may exceed its current allocation based on the weighting values associated with each partition. In other words, excess bandwidth, in one implementation is distributed based on the relative weights of the child partitions. FIG. 7 illustrates the overall process flow associated with one implementation of the present invention. At periodic intervals (380), the aggregate bandwidth allocation daemon operates to invoke a process that, as discussed below, re-

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	computes aggregate bandwidth allocations across the partitions (382), and a process (discussed above) to re-compute the per-flow bandwidth allocations within the partitions (384).” <i>See Feroz at claim 1.</i>
[1c] determining, by the network switching element, a fair-share bandwidth allocation for the target port, the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and	Feroz discloses determining, by the network switching element, a fair-share bandwidth allocation for the target port, the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element. <i>See Feroz at Abstract:</i> “Methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates.” <i>See Feroz 3:9-54:</i> “To facilitate monitoring, management and control of network environments, a variety of network devices, applications, technologies and services have been developed. For example, certain data flow rate control mechanisms have been developed to provide a means to control and optimize efficiency of data transfer as well as allocate available bandwidth among a variety of business enterprise functionalities. For example, U.S. Pat. No. 6,038,216 discloses a method for explicit data rate control in a packet-based network environment without data rate supervision. Data rate control directly moderates the rate of data transmission from a sending host, resulting in just-in-time data transmission to control inbound traffic and buffering of packets, and reduce the inefficiencies associated with dropped packets. Bandwidth management devices also allow for explicit data rate control for flows associated with a particular traffic classification. For example, U.S. Pat. No. 6,412,000, above, discloses automatic classification of network traffic for use in connection with bandwidth allocation mechanisms. U.S. Pat. No. 6,046,980 discloses systems and methods allowing for application layer control of bandwidth utilization in packet-based computer networks. For example, bandwidth management devices allow network administrators to specify policies operative to control and/or prioritize the bandwidth allocated to individual data flows according to

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	traffic classifications. In addition, certain bandwidth management devices, as well as certain routers, allow network administrators to specify aggregate bandwidth utilization controls to divide available bandwidth into partitions. With some network devices, these partitions can be configured to provide a minimum bandwidth guarantee, and/or cap bandwidth, as to a particular class of traffic. An administrator specifies a traffic class (such as FTP data, or data flows involving a specific user or network application) and the size of the reserved virtual link-i.e., minimum guaranteed bandwidth and/or maximum bandwidth. Such partitions can be applied on a per-application basis (protecting and/or capping bandwidth for all traffic associated with an application) or a per-user basis (controlling, prioritizing, protecting and/or capping bandwidth for a particular user). In addition, certain bandwidth management devices allow administrators to define a partition hierarchy by configuring one or more partitions dividing the access Link and further dividing the parent partitions into one or more child partitions. U.S. patent application Ser. No. 10/108,085 discloses data structures and methods for implementing a partition hierarchy.” <i>See Feroz at 4:17-36:</i> “While partitions and per-flow rate policies are effective for their intended purposes, the configuration parameters that lead to their effective and predictable concurrent operation is often beyond the skill level of the average end user. For example, effective configuration of per-flow rate policies within a partition often requires a general knowledge of the typical number of data flows falling within the partition at given times. For example, a per-flow target rate is generally allocated based on an estimated rate demand for a given flow, regardless of the configuration of the partition that controls the flows in the aggregate. This per-flow target rate may be limited or adjusted by a per-flow rate policy configured by a user. In either case, these circumstances can result in significant under utilization of available bandwidth if the data flows do not actually consume their allocated rate, resulting in under utilization of the bandwidth allocated to the partition and the access link itself. Furthermore, use of explicit per-flow rate policies as the primary means of allocating per-flow bandwidth may not achieve a fair distribution of this bandwidth across the active flows in the partition.” <i>See Feroz at 4:46-61:</i> “The present invention provides methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates. These and other aspects of various implementations of the present invention will become apparent from the drawings and description provided below.” <i>See Feroz at 6:4-25:</i> “As FIG. 2 illustrates, network application traffic management device 130, in one implementation, comprises network device application processor 75, and first and second network interfaces 71, 72, which operably connect application traffic management device 130 to the communications path between router 22 and network 40. Network device application processor 75 generally refers to the functionality implemented by application traffic management device 130, such as network monitoring or reporting, application traffic management, and the like. In one embodiment, network device application processor 75 is a combination of hardware and software, such as a central processing unit, memory, a system bus, an operating system, device drivers, and one or more software modules implementing the functions performed by application traffic management device 130. For didactic purposes, application traffic management device 130 is configured to manage network traffic traversing access link 21. The above-identified patents and patent applications, incorporated by reference herein, disclose various functionalities and features that may be incorporated into application traffic management devices according to various implementations of the present invention.” <i>See Feroz at 6:26-50:</i> “In one embodiment, first and second network interfaces 71, 72 are the hardware communications interfaces that receive and transmit packets over the computer network environment. In one implementation, first and second network interfaces 71, 72 reside on separate network interface cards operably connected to the system bus of application traffic management device 130. In another implementation, first and second network interfaces reside on the same network interface card. In addition, the first and second network interfaces 71, 72 can be wired network interfaces, such as Ethernet (IEEE 802.3) interfaces, and/or wireless network interfaces, such as IEEE 802.11, BlueTooth, satellite-based interfaces, and the like. As FIG. 2 illustrates, application traffic management device 130, in one embodiment, includes persistent memory 76, such as a hard disk drive or other suitable memory device, such as writable CD, DVD, or tape drives. In other implementations, application traffic management device 130 can include additional network interfaces, beyond network interfaces 71 and 72, to support additional access links or other

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	functionality. Furthermore, U.S. application Ser. No. 10/843,185 provides a description of the operation of various modules (according to one possible implementation of the present invention), such as network interface drivers, and data structures for receiving into memory and processing packets encountered at network interfaces 71, 72.” <i>See Feroz at 6:51-7:9:</i> “As FIG. 2 illustrates, network device application processor 75, in one implementation, includes a packet processor 92, flow control module 94, and traffic classification engine 96. Network device application processor 75, in one implementation, further comprises host database 134, flow database 135, measurement engine 140, management information base 138, and administrator interface 150. In one embodiment, the packet processor 92 is operative to process data packets, such as detecting new data flows, parsing the data packets for various attributes (such by a network administrator, in a hierarchical traffic class structure. In one implementation, flow control module 94 is operative to apply aggregate and per-flow bandwidth utilization controls to data flows traversing the access link 21 in the inbound and/or outbound directions.” <i>See Feroz at 9:38-45:</i> “As discussed above, traffic classification engine 96, in one implementation, is operative to classify data flows into one of a plurality of traffic classes. Traffic classification engine 96, in one implementation, comprises a plurality of service type identification modules, each of which correspond to a set of service types. Each service type identification module analyzes one or more packets in a given data flow to attempt to identify a service type corresponding to the flow.” <i>See Feroz at Feroz at 11:13-23:</i> “A traffic class comprises a set of matching rules or attributes allowing for logical grouping of data flows that share the same characteristic or set of characteristics. In one implementation, the matching rules can correspond to the service type identifiers discussed above, as well as other data flow attributes, such as the network interface on which the packets are received by application traffic management device 130, whether the server is the inside or outside host (see above), non-standard and standard port numbers, host IP address or subnet, MAC address, application-specific strings, diffserv codes, MPLS tags, VLAN tags, and the like.” <i>See Feroz at 12:8-13:</i> “Traffic classification engine 96, in one implementation, stores traffic classes associated with data flows that traverse access link 21. Traffic classification engine 96, in one embodiment, stores the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	traffic classes and corresponding data (e.g., matching rules, policies, partition pointers, etc.) related to each traffic class in a hierarchical tree.” <i>See Feroz at 13:7-38:</i> “As discussed more fully below, flow control module 94 enforces aggregate, partition-based bandwidth utilization controls on data flows traversing access link 21. In one implementation, flow control module 94 monitors bandwidth demand of each partition relative to an aggregate, weighted fair share allocation scheme, and dynamically re-distributes unutilized bandwidth to partitions whose demand for bandwidth exceeds its weighted fair share allocation. As discussed in more detail below, flow control module 94 performs these dynamic allocation adjustments on a continuous basis, as needed. In one implementation, the demand for bandwidth at a given partition can be evaluated in a number of ways. For example, the size of, and rate at which, packets in a flow are received by application traffic management device 130 during a given time interval can be monitored to compute a per-flow rate demand. The per-flow rate demands for all flows falling within a given partition can be aggregated to yield an aggregate rate demand for the partition. In addition, these computed rate values can be smoothed by maintaining a rolling average or weighted moving average across a sliding window of time. Aggregate rate demand for a partition may also be determined by analyzing the number of packets queued in a corresponding partition queue over a given analysis interval. For example, the change in the number of packets queued in a given partition queue over an analysis interval can be analyzed to determine the aggregate rate demand for a given partition. As discussed in more detail below, flow control module 94 can then dynamically allocate unutilized bandwidth to partitions where the aggregate rate demand exceeds their respective bandwidth allocations. In one implementation, the weighting values associated with the partitions determines in part how unutilized bandwidth is distributed.” <i>See Feroz at 13:39-64:</i> “In addition, flow control module 94, in some implementations, can optionally apply per-flow bandwidth utilization controls, on data flows traversing access link 21. In one implementation, flow control module 94 includes a partitioning module operative to enforce aggregate bandwidth utilization controls (e.g., partitions), and a per-flow rate control module operative to apply per-flow rate controls on data flows. In one implementation, flow control module 94 implements the deterministic partition scheduling functionality disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference herein. Furthermore, as discussed more fully below, flow control module 94 includes functionality for dynamically adjusting aggregate bandwidth allocation among partitions. In one implementation, flow control module 94 also includes processes that dynamically

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	adjust per-flow bandwidth allocation within partitions. A partition operates to manage bandwidth for aggregate data flows associated with a traffic class. As discussed above, a partition is essentially a division of the capacity of access link 21. Still further, flow control module 94, in some implementations, may also implement per-flow rate policies on the data flows. In one implementation, flow control module 94 implements the TCP Rate Control technologies disclosed in U.S. Pat. No. 6,038,216 to control the rate at which transmitters send data and therefore the amount of data that is queued in buffers at application traffic management device 130.” <i>See Feroz at 14:13-20:</i> “Similar to the traffic class tree, the root partition node represents the capacity of the access link in either the inbound or outbound directions. In one implementation, separate partition configurations and controls can be maintained for data flows traveling in the inbound or outbound directions relative to network 40. In other implementations, a single partition configuration can be applied to both inbound and outbound data flows.” <i>See Feroz at 14:21-41:</i> “Once a partition configuration is established, in one implementation, a user can simply assign a weight to each partition node, which essentially expresses the priority of the network traffic class(es) falling within the partition relative to other traffic classes. Optionally, a user may also assign a minimum aggregate bandwidth allocation to one or more of the selected partition nodes. As discussed in more detail below, each partition is then allocated a fair share of available bandwidth based on the weights assigned to the partitions, limited by the minimum and maximum bandwidth settings configured for any partitions. In another implementation, a user may also specify a maximum bandwidth allocation for one or more partitions. In one implementation, unless the user specifically configures a weighting value for at least one partition in a given level of the hierarchy, by default, each sibling partition is weighted equally. Additionally, if a user configures a weighting value for only one partition, each sibling partition is configured with a default weighting value. In one implementation, administrator interface 150 requires the user to enter weighting values for all sibling partitions after a user initially configures a weighting value for a selected partition.” <i>See Feroz at 14:42-58:</i> “After the user has completed a configuration, administrator interface 150, in one implementation, processes the configuration to create an aggregate bandwidth allocation scheme that includes partitions, and optionally per-flow rate policies, based on the received configuration. For example,

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	partition queues are created for each leaf partition, and weighted, fair share bandwidth allocations are computed for the partitions, as discussed in more detail below. In addition, in one implementation, each active data flow within a given partition is also guaranteed a fair share of the bandwidth allocated to the partition. The aggregate fair share allocation for each partition is generally stable, unless there are configuration changes. However, the fair share allocation for each flow changes dynamically based on the number of active flows in the partition. In addition, the actual bandwidth allocated to a given partition at a given point in time may change dynamically in response to network traffic demands” See Feroz at 15:6-14: $\text{Aggregate Weight} = \sum \text{Weights from Child Nodes}$ $\text{Allocation Per Partition } (c) = \frac{\text{Parent Allocation} * \text{Configured Weight of Partition } (c)}{\text{Aggregate Weight}}$ See Feroz at 16:56-17:17: “FIG. 6 provides, for didactic purposes, an exemplary hierarchical partition configuration. Root partition node 82 represents the total capacity of access link 21 in either the outbound or inbound direction. In one implementation, separate data structures and processes are maintained for the outbound and inbound directions. Partition nodes 83 represent a first set of partitions allocating the capacity of access link at a first level. As FIG. 6 shows, each parent partition may include child partitions represented as partition nodes 83 and 84. For didactic purposes, assume that access link 21 is a T1 line and, thus, provides maximum available bandwidth of 1.5 Mbps. Partitions P1, P2, and P3 may be configured with weights, as discussed above, to equally divide access link 21 into 0.5 Mbps partitions at a first level. In addition, child partitions P4 and P5 may further divide parent partition P1 into a 0.2 Mbps partition (P4) and a 0.3 Mbps partition (P5). Similarly, child partitions P6, P7 and P8 may divide parent partition P2 into two 0.1 Mbps partitions (P6 and P8), and a 0.3 Mbps partition (P7). As discussed above, the initial aggregate bandwidth allocations may also be

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	derived from weights associated with the partition nodes. Of course, any suitable partition configuration may be employed, including the configurations illustrated in FIGS. 9A thru 9F, See above. Still further, other configurations are possible. For example, the root node may represent the access link in both the inbound and outbound direction, while child partitions from the root can correspond to traffic encountered at a given network interface (such as interface 71) and destined for egress from a second network interface (such as interface 72).” <i>See Feroz at 20:6-20:</i> “FIG. 4B sets forth a method, according to one implementation of the invention, directed to computing a per-flow fair share allocation and a target rate for flows within a given partition. As FIG. 4B illustrates, the per-flow fair share bandwidth allocation is the current aggregate bandwidth allocated to the partition divided by the number of active flows (330). As discussed above, the current number of active flows is maintained in a traffic class or partition object. The per-flow fair share rate can be used in a variety of ways. In one implementation, the per-flow fair share rate can be used as the basis of determining whether packets corresponding to a flow are placed on the high or low priority partition queue. In another implementation, the per-flow fair share rate can be used in a process that computes a target rate for each flow within a partition.” <i>See Feroz at 20:46-21:15:</i> “As FIG. 4B illustrates, flow control module 94, in one implementation, distributes some of the unutilized bandwidth within a partition to flows having compensation demand in the partition. In one implementation, flow control module 94, for all active flows in the partition (332), checks whether the current rate of the flow is less than X percent of the target rate of the flow (336). As discussed above, in one implementation, X=94. As FIG. 4B illustrates, the target rate for new flows (333) is the per-flow fair share previously computed (330). If the current rate of the flow is less than X percent of the target rate, flow control module 94 then determines whether a flow compensation demand has been previously allocated to the flow in a previous cycle (344). If so, flow control module 94 then determines whether the current rate is less than Y percent of the target rate (346). In one implementation, Y equals 88; however, other values can be used depending on engineering and design considerations. If both 344 and 346 are met, flow control module 94 steps the flow compensation demand (FCD) down by Z percent of the previously allocated flow compensation demand (348). In one implementation, Z equals 6 percent; however, other values can be used. While other implementations are contemplated by the invention, the foregoing prevents temporary fluctuations in data rates from substantially affecting the flow target rate, thereby allowing the rate of the flow to potentially increase. Otherwise, if a flow compensation demand was not previously

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	allocated to the flow (344), flow control module 94 assigns a target rate equal to the lesser of 1) the current rate of the flow multiplied by a factor M, which in one implementation is 1.5; or 2) the per-flow fair share bandwidth allocation (349). As FIG. 4B illustrates, the excess bandwidth (Fair Share less Current Rate), if any, is passed up to the partition to enable subsequent calculations of aggregate bandwidth underutilization. Assigning a target rate that is greater than the measured current rate helps to avoid the recursive spiral by which the flow data rate decreases, allowing the flow data rate to increase up to the fair share allocation.” <i>See Feroz at 22:4-24:</i> “In one implementation, flow control module 94 includes an aggregate bandwidth allocation daemon that operates at periodic intervals (e.g., 125 milliseconds, etc.) to update the aggregate bandwidth allocation across the partitions based on the weighted fair share allocations, and the observed aggregate compensation demand and aggregate bandwidth underutilization. In one implementation, flow control module 94 distributes unutilized or excess bandwidth to partitions whose demand for bandwidth may exceed its current allocation based on the weighting values associated with each partition. In other words, excess bandwidth, in one implementation is distributed based on the relative weights of the child partitions. FIG. 7 illustrates the overall process flow associated with one implementation of the present invention. At periodic intervals (380), the aggregate bandwidth allocation daemon operates to invoke a process that, as discussed below, re-computes aggregate bandwidth allocations across the partitions (382), and a process (discussed above) to re-compute the per-flow bandwidth allocations within the partitions (384).” <i>See also Feroz Figs. 8A and 8B and associated discussion at 22:24-24:59.</i> <i>See Feroz at claim 1.</i>
[1d] adjusting, by the network switching element, the bandwidth allocation for the target port based on the fair-share bandwidth allocation.	Feroz discloses adjusting, by the network switching element, the bandwidth allocation for the target port based on the fair-share bandwidth allocation. <i>See Feroz at Abstract:</i> “Methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates.” <i>See Feroz at 4:46-61:</i> “The present invention provides methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates. These and other aspects of various implementations of the present invention will become apparent from the drawings and description provided below.” <i>See Feroz at 6:51-7:9:</i> “As FIG. 2 illustrates, network device application processor 75, in one implementation, includes a packet processor 92, flow control module 94, and traffic classification engine 96. Network device application processor 75, in one implementation, further comprises host database 134, flow database 135, measurement engine 140, management information base 138, and administrator interface 150. In one embodiment, the packet processor 92 is operative to process data packets, such as detecting new data flows, parsing the data packets for various attributes (such by a network administrator, in a hierarchical traffic class structure. In one implementation, flow control module 94 is operative to apply aggregate and per-flow bandwidth utilization controls to data flows traversing the access link 21 in the inbound and/or outbound directions.” <i>See Feroz at 7:10-28:</i> “As discussed above, in one implementation, network device application processor 75 further comprises measurement engine 140, management information base (MIB) 138, and administrator interface 150. Management information base 138 is a database of standard and extended network objects related to the operation of application traffic management device 130. Measurement engine 140 maintains measurement and statistical data relating to operation of application traffic management device 130 to allow for monitoring of bandwidth utilization and network performance across access Link 21 with respect to a plurality of bandwidth utilization and other network

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	statistics on an aggregate and/or per-traffic-class level. In one implementation, measurement engine 140 tracks a variety of metrics corresponding to the partitions implemented by flow control module 94. These metrics allow flow control module 94, as discussed in more detail below, to dynamically adjust bandwidth allocations across flows and partitions to improve or manage network application performance.” <i>See Feroz at 7:43-48:</i> “As discussed above, packet processor 92, in one implementation, is operative to detect new data flows, instantiate data structures associated with the flows and parse packets to identify packet attributes, such as source and destination addresses, port numbers, etc., and populate one or more fields in the data structures. . . .” <i>See Feroz at 8:6-12:</i> “As discussed more fully below, in one embodiment, flows are identified based on the following flow attributes: 1) source IP address, 2) destination IP address, 3) source port number, 4) destination port number, and 5) protocol (derived from the “protocol” field in IPv4 headers, and the “NextHeader” field in IPv6 headers).” <i>See Feroz at 8:17-42:</i> “For a TCP flow, for example, packet processor 92 can determine a new data flow by detecting SYN, SYN/ACK, and/or ACK packets. . . . The termination of TCP connections is typically detected by identifying FIN packets; however, the timeout mechanisms discussed above can be used in situations where a FIN packet is not detected.” <i>See Feroz at 8:61-9:9:</i> “For a TCP/IP packet, packet processor 92 can compute the inside and outside addresses based on the source and destination network addresses of the packet and the direction of the packet flow. Still further, packet processor 92 can also identify which host is the client and which host is the server for a given data flow and store this information in the flow specification or flow object. The identification of a server or client in a given transaction generally depends on the network protocols employed by the hosts. For example, in TCP flows, a client initiates a transaction by transmitting a SYN packet to initiate a TCP connection. Application traffic management device 130 can detect the SYN packet and note the source network address of the packet as the client host, and the destination

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	address as the server host. One of ordinary skill in the art will recognize how to identify clients and servers in connection with other networking protocols.” <i>See Feroz at 9:10-36:</i> “In one embodiment, packet processor 92 creates and stores flow objects corresponding to data flows in flow database 135. In one embodiment, flow object attributes include a pointer to a corresponding flow specification object, as well as other flow state parameters, such as TCP connection status, timing of last packets in the inbound and outbound directions, speed information, apparent round trip time, packet count, etc. Flow object attributes further include at least one traffic class identifier (or pointer(s) thereto) associated with the data flow, as well as policy parameters (or pointers thereto) corresponding to the identified traffic class. In one embodiment, flow objects further include a list of traffic classes for which measurement data (maintained by measurement engine 140) associated with the data flow should be logged. In one embodiment, to facilitate association of an existing flow object to subsequent packets associated with a data flow or connection, flow database 135 further maintains a control block hash table including a key comprising a hashed value computed from a string comprising the inside IP address, outside IP address, inside port number, outside port number, and protocol type (e.g., TCP, UDP, etc.) associated with a pointer to the corresponding flow object. According to this embodiment, to identify whether a flow object exists for a given data flow, packet processor 92 hashes the values identified above and scans the hash table for a matching entry. If one exists, packet processor 92 associates the pointer to the corresponding flow object with the packets in the data flow.” <i>See Feroz at 9:38-45:</i> “As discussed above, traffic classification engine 96, in one implementation, is operative to classify data flows into one of a plurality of traffic classes. Traffic classification engine 96, in one implementation, comprises a plurality of service type identification modules, each of which correspond to a set of service types. Each service type identification module analyzes one or more packets in a given data flow to attempt to identify a service type corresponding to the flow.” <i>See Feroz at Feroz at 11:13-23:</i> “A traffic class comprises a set of matching rules or attributes allowing for logical grouping of data flows that share the same characteristic or set of characteristics. In one implementation, the matching rules can correspond to the service type identifiers discussed above, as well as other data flow attributes, such as the network interface on which the packets are received by application traffic management device 130, whether the server is the inside or outside host (see above), non-

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	standard and standard port numbers, host IP address or subnet, MAC address, application-specific strings, diffserv codes, MPLS tags, VLAN tags, and the like.” <i>See Feroz at 12:8-13:</i> “Traffic classification engine 96, in one implementation, stores traffic classes associated with data flows that traverse access link 21. Traffic classification engine 96, in one embodiment, stores the traffic classes and corresponding data (e.g., matching rules, policies, partition pointers, etc.) related to each traffic class in a hierarchical tree.” <i>See Feroz at 13:7-38:</i> “As discussed more fully below, flow control module 94 enforces aggregate, partition-based bandwidth utilization controls on data flows traversing access link 21. In one implementation, flow control module 94 monitors bandwidth demand of each partition relative to an aggregate, weighted fair share allocation scheme, and dynamically re-distributes unutilized bandwidth to partitions whose demand for bandwidth exceeds its weighted fair share allocation. As discussed in more detail below, flow control module 94 performs these dynamic allocation adjustments on a continuous basis, as needed. In one implementation, the demand for bandwidth at a given partition can be evaluated in a number of ways. For example, the size of, and rate at which, packets in a flow are received by application traffic management device 130 during a given time interval can be monitored to compute a per-flow rate demand. The per-flow rate demands for all flows falling within a given partition can be aggregated to yield an aggregate rate demand for the partition. In addition, these computed rate values can be smoothed by maintaining a rolling average or weighted moving average across a sliding window of time. Aggregate rate demand for a partition may also be determined by analyzing the number of packets queued in a corresponding partition queue over a given analysis interval. For example, the change in the number of packets queued in a given partition queue over an analysis interval can be analyzed to determine the aggregate rate demand for a given partition. As discussed in more detail below, flow control module 94 can then dynamically allocate unutilized bandwidth to partitions where the aggregate rate demand exceeds their respective bandwidth allocations. In one implementation, the weighting values associated with the partitions determines in part how unutilized bandwidth is distributed.” <i>See Feroz at 13:39-64:</i> “In addition, flow control module 94, in some implementations, can optionally apply per-flow bandwidth utilization controls, on data flows traversing access link 21. In one implementation, flow control module 94 includes a partitioning module operative to enforce aggregate bandwidth

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	utilization controls (e.g., partitions), and a per-flow rate control module operative to apply per-flow rate controls on data flows. In one implementation, flow control module 94 implements the deterministic partition scheduling functionality disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference herein. Furthermore, as discussed more fully below, flow control module 94 includes functionality for dynamically adjusting aggregate bandwidth allocation among partitions. In one implementation, flow control module 94 also includes processes that dynamically adjust per-flow bandwidth allocation within partitions. A partition operates to manage bandwidth for aggregate data flows associated with a traffic class. As discussed above, a partition is essentially a division of the capacity of access link 21. Still further, flow control module 94, in some implementations, may also implement per-flow rate policies on the data flows. In one implementation, flow control module 94 implements the TCP Rate Control technologies disclosed in U.S. Pat. No. 6,038,216 to control the rate at which transmitters send data and therefore the amount of data that is queued in buffers at application traffic management device 130.” <i>See Feroz at 14:13-20:</i> Incorporated by reference U.S. Pat. No. 6,038,216 discloses “the window size is set to allow the specified rate given the current round trip time ("RTT") and the detected speed.” “Similar to the traffic class tree, the root partition node represents the capacity of the access link in either the inbound or outbound directions. In one implementation, separate partition configurations and controls can be maintained for data flows traveling in the inbound or outbound directions relative to network 40. In other implementations, a single partition configuration can be applied to both inbound and outbound data flows.” <i>See Feroz at 14:21-41:</i> “Once a partition configuration is established, in one implementation, a user can simply assign a weight to each partition node, which essentially expresses the priority of the network traffic class(es) falling within the partition relative to other traffic classes. Optionally, a user may also assign a minimum aggregate bandwidth allocation to one or more of the selected partition nodes. As discussed in more detail below, each partition is then allocated a fair share of available bandwidth based on the weights assigned to the partitions, limited by the minimum and maximum bandwidth settings configured for any partitions. In another implementation, a user may also specify a maximum bandwidth allocation for one or more partitions. In one implementation, unless the user specifically configures a weighting value for at least one partition in a given level of the hierarchy, by default, each sibling partition is weighted equally. Additionally, if a user configures a weighting value for only one partition, each sibling partition is configured with a default weighting

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	value. In one implementation, administrator interface 150 requires the user to enter weighting values for all sibling partitions after a user initially configures a weighting value for a selected partition.” <i>See Feroz at 14:42-58:</i> “After the user has completed a configuration, administrator interface 150, in one implementation, processes the configuration to create an aggregate bandwidth allocation scheme that includes partitions, and optionally per-flow rate policies, based on the received configuration. For example, partition queues are created for each leaf partition, and weighted, fair share bandwidth allocations are computed for the partitions, as discussed in more detail below. In addition, in one implementation, each active data flow within a given partition is also guaranteed a fair share of the bandwidth allocated to the partition. The aggregate fair share allocation for each partition is generally stable, unless there are configuration changes. However, the fair share allocation for each flow changes dynamically based on the number of active flows in the partition. In addition, the actual bandwidth allocated to a given partition at a given point in time may change dynamically in response to network traffic demands” <i>See Feroz at 15:6-14:</i> $\text{Aggregate Weight} = \sum \text{Weights from Child Nodes}$ $\text{Allocation Per Partition } (c) = \frac{\text{Parent Allocation} * \text{Configured Weight of Partition } (c)}{\text{Aggregate Weight}}$ <i>See Feroz at 16:56-17:17:</i> “FIG. 6 provides, for didactic purposes, an exemplary hierarchical partition configuration. Root partition node 82 represents the total capacity of access link 21 in either the outbound or inbound direction. In one implementation, separate data structures and processes are maintained for the outbound and inbound directions. Partition nodes 83 represent a first set of partitions allocating the capacity of access link at a first level. As FIG. 6 shows, each parent partition may include child

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	partitions represented as partition nodes 83 and 84. For didactic purposes, assume that access link 21 is a T1 line and, thus, provides maximum available bandwidth of 1.5 Mbps. Partitions P1, P2, and P3 may be configured with weights, as discussed above, to equally divide access link 21 into 0.5 Mbps partitions at a first level. In addition, child partitions P4 and P5 may further divide parent partition P1 into a 0.2 Mbps partition (P4) and a 0.3 Mbps partition (P5). Similarly, child partitions P6, P7 and P8 may divide parent partition P2 into two 0.1 Mbps partitions (P6 and P8), and a 0.3 Mbps partition (P7). As discussed above, the initial aggregate bandwidth allocations may also be derived from weights associated with the partition nodes. Of course, any suitable partition configuration may be employed, including the configurations illustrated in FIGS. 9A thru 9F, See above. Still further, other configurations are possible. For example, the root node may represent the access link in both the inbound and outbound direction, while child partitions from the root can correspond to traffic encountered at a given network interface (such as interface 71) and destined for egress from a second network interface (such as interface 72). . . .” <i>See Feroz at 18:28-62:</i> “FIG. 3 illustrates, packet processor 92 updates, for existing data flows, attributes of the flow object in response to the packet such as the packet count, last packet time, and the like (208). Packet processor 92 can also perform other operations, such as analyzing the packets for connection state information. For example, packet processor 92 can inspect various TCP flags to determine whether the received packet is part of a new data flow or represents a change to an existing data flow (such as the first data packet after the TCP handshake). Methods for determining new data flows and assigning packets to existing data flows are well known in the art and also depend on the particular transport layer protocol employed. For a TCP packet, packet processor 92 can determine a new data flow by detecting SYN and/or SYN/ACK packets. However, a new data flow can simply be a data flow for which there is no corresponding flow object in flow database 135. . . . In one implementation, this separate process can also be passed pointers to TCP FIN packets to adjust various data structures, such as reallocating flow object space to a pool of available memory, adjusting flow counts in traffic class or partition objects.” <i>See Feroz at 19:12-26:</i> “For example, each partition object, in one implementation, may include the following statistics: 1) a flow count (the number of currently active flows); 2) current rate (a weighted moving average in bps of the rate observed for the flows in the class); 3) aggregate compensation demand; and 4) aggregate unutilized fair share. In addition, per-flow statistics are also maintained, such as the current rate of the flow (a weighted moving average over an analysis interval of the rate at which

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	packets are transmitted from application traffic management device), packet count, and the like. One skilled in the art will recognize that the order of processing operations illustrated in FIG. 3 is one possible process flow and that the flow control processes described herein can be performed at other points in the packet processing path.” <i>See Feroz at 19:28-20:4:</i> “FIG. 4A illustrates a process, according to one implementation of the invention, directed to reading packets onto flow and partition queues. . . . In addition, since the packet represents the addition of a new flow to a partition, flow control module 94 re-computes the per-flow bandwidth allocation for the partition (316), as discussed in a following section. . . . In one implementation, if the initial flow rate of the flow is less than the per-flow fair share for the partition (318), flow control module 94 pushes the flow queue onto the high priority queue (322), facilitating the chance for the rate of the flow to increase. Otherwise, flow control module 94 pushes the flow queue onto the low priority queue 60 a corresponding to the identified partition (320). In one implementation, flow control module 94 applies a process directed to the per-flow rate enforcement. In one implementation, the per-flow target rate is enforced by mapping the target rate to an advertised TCP window size, as disclosed in U.S. Pat. No. 6,038,216, and modifying the advertised window size in the TCP ACK fields of packets transmitted back to the transmitter to modify the rate at which subsequent packets are sent. Of course, the per-flow rate control processes may be applied to the packets at any point prior to egress from application traffic management device 130. In addition, the use of TCP Rate Control technologies to control the rate of data transmission from end hosts is an optional feature.” <i>See Feroz at 20:6-20:</i> “FIG. 4B sets forth a method, according to one implementation of the invention, directed to computing a per-flow fair share allocation and a target rate for flows within a given partition. As FIG. 4B illustrates, the per-flow fair share bandwidth allocation is the current aggregate bandwidth allocated to the partition divided by the number of active flows (330). As discussed above, the current number of active flows is maintained in a traffic class or partition object. The per-flow fair share rate can be used in a variety of ways. In one implementation, the per-flow fair share rate can be used as the basis of determining whether packets corresponding to a flow are placed on the high or low priority partition queue. In another implementation, the per-flow fair share rate can be used in a process that computes a target rate for each flow within a partition.”

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	See Feroz at 20:21-45 “According to one implementation, the allocation of bandwidth to flows within a given partition is an optional step. In such an implementation, as FIG. 4B illustrates, flow control module 94 computes the aggregate compensation demand and the aggregate bandwidth underutilization for the partition (331). Compensation demand occurs when the current rate of a flow exceeds a threshold based on a percentage of the target rate of the flow. In one implementation, the compensation demand threshold is 94% of the per-flow target rate; however, this percentage can vary based on various engineering and design considerations. The compensation demand for a flow, in one implementation, is a fixed percentage of the detected current rate for the flow. As discussed above, the current rate can be estimated, in one implementation, by monitoring the size of incoming packets and the time at which they are received. Aggregate compensation demand for a partition is the sum of the compensation demands for each flow. Oppositely, bandwidth underutilization (unutilized fair share) occurs when the current rate associated with the flow is less than a percentage of the target rate, and is computed based on the difference between the per-flow fair share bandwidth allocation and the current rate. In one implementation, the aggregate bandwidth underutilization for a partition is thus the sum of the individual bandwidth underutilization values for the flows in the partition.” See Feroz at 20:46-21:15: “As FIG. 4B illustrates, flow control module 94, in one implementation, distributes some of the unutilized bandwidth within a partition to flows having compensation demand in the partition. In one implementation, flow control module 94, for all active flows in the partition (332), checks whether the current rate of the flow is less than X percent of the target rate of the flow (336). As discussed above, in one implementation, X=94. As FIG. 4B illustrates, the target rate for new flows (333) is the per-flow fair share previously computed (330). If the current rate of the flow is less than X percent of the target rate, flow control module 94 then determines whether a flow compensation demand has been previously allocated to the flow in a previous cycle (344). If so, flow control module 94 then determines whether the current rate is less than Y percent of the target rate (346). In one implementation, Y equals 88; however, other values can be used depending on engineering and design considerations. If both 344 and 346 are met, flow control module 94 steps the flow compensation demand (FCD) down by Z percent of the previously allocated flow compensation demand (348). In one implementation, Z equals 6 percent; however, other values can be used. While other implementations are contemplated by the invention, the foregoing prevents temporary fluctuations in data rates from substantially affecting the flow target rate, thereby allowing the rate

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	of the flow to potentially increase. Otherwise, if a flow compensation demand was not previously allocated to the flow (344), flow control module 94 assigns a target rate equal to the lesser of 1) the current rate of the flow multiplied by a factor M, which in one implementation is 1.5; or 2) the per-flow fair share bandwidth allocation (349). As FIG. 4B illustrates, the excess bandwidth (Fair Share less Current Rate), if any, is passed up to the partition to enable subsequent calculations of aggregate bandwidth underutilization. Assigning a target rate that is greater than the measured current rate helps to avoid the recursive spiral by which the flow data rate decreases, allowing the flow data rate to increase up to the fair share allocation.” <i>See Feroz 21:29-31:</i> “For example, the target rate can be simply set to the per-flow fair share computed for the partition regardless of the current rate.” <i>See Feroz at 22:4-24:</i> “In one implementation, flow control module 94 includes an aggregate bandwidth allocation daemon that operates at periodic intervals (e.g., 125 milliseconds, etc.) to update the aggregate bandwidth allocation across the partitions based on the weighted fair share allocations, and the observed aggregate compensation demand and aggregate bandwidth underutilization. In one implementation, flow control module 94 distributes unutilized or excess bandwidth to partitions whose demand for bandwidth may exceed its current allocation based on the weighting values associated with each partition. In other words, excess bandwidth, in one implementation is distributed based on the relative weights of the child partitions. FIG. 7 illustrates the overall process flow associated with one implementation of the present invention. At periodic intervals (380), the aggregate bandwidth allocation daemon operates to invoke a process that, as discussed below, re-computes aggregate bandwidth allocations across the partitions (382), and a process (discussed above) to re-compute the per-flow bandwidth allocations within the partitions (384).” <i>See Feroz at claim 1.</i>
[11pre] A method of adjusting bandwidth allocation by a network controller in a communications network, the network controller being	To the extent the preamble is limiting, Feroz discloses this preamble. <i>See claim element [1pre] supra.</i>

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
configured to control a network switching element, the network switching element including a target port, the method comprising:	
[11a] receiving, by the network controller, data flow information from the network switching element, the data flow information including information about data flows traversing of the target port of the network switching element;	<i>See claim element [1a] supra.</i>
[11b] determining, by the network controller, a bandwidth allocation of the target port based on the data flow information, the bandwidth allocation of the target port being a bandwidth that is currently allocated for each of the data flows traversing the target port;	<i>See claim element [1b] supra.</i>
[11c] determining, by the network controller, an over-subscription ratio, the over-subscription ratio being a ratio of the bandwidth allocation of the target port to a number of data flows traversing the target part;	Feroz discloses determining, by the network controller, an over-subscription ratio, the over-subscription ratio being a ratio of the bandwidth allocation of the target port to a number of data flows traversing the target part. <i>See claim element [1b] supra.</i> <i>See Feroz at Abstract:</i> “Methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates.” <i>See Feroz at 4:17-36:</i> “While partitions and per-flow rate policies are effective for their intended purposes, the configuration parameters that lead to their effective and predictable concurrent operation is often beyond the skill level of the average end user. For example, effective configuration of per-flow rate policies within a partition often requires a general knowledge of the typical number of data flows falling within the partition at given times. For example, a per-flow target rate is generally allocated based on an estimated rate demand for a given flow, regardless of the configuration of the partition that controls the flows in the aggregate. This per-flow target rate may be limited or adjusted by a per-flow rate policy configured by a user. In either case, these circumstances can result in significant under utilization of available bandwidth if the data flows do not actually consume their allocated rate, resulting in under utilization of the bandwidth allocated to the partition and the access link itself. Furthermore, use of explicit per-flow rate policies as the primary means of allocating per-flow bandwidth may not achieve a fair distribution of this bandwidth across the active flows in the partition.” <i>See Feroz at 4:46-61:</i> “The present invention provides methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates. These and other aspects of various implementations of the present invention will become apparent from the drawings and description provided below.” <i>See Feroz at 6:51-7:9:</i> “As FIG. 2 illustrates, network device application processor 75, in one implementation, includes a

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	packet processor 92, flow control module 94, and traffic classification engine 96. Network device application processor 75, in one implementation, further comprises host database 134, flow database 135, measurement engine 140, management information base 138, and administrator interface 150. In one embodiment, the packet processor 92 is operative to process data packets, such as detecting new data flows, parsing the data packets for various attributes (such by a network administrator, in a hierarchical traffic class structure. In one implementation, flow control module 94 is operative to apply aggregate and per-flow bandwidth utilization controls to data flows traversing the access link 21 in the inbound and/or outbound directions.” <i>See Feroz at 7:43-48:</i> “As discussed above, packet processor 92, in one implementation, is operative to detect new data flows, instantiate data structures associated with the flows and parse packets to identify packet attributes, such as source and destination addresses, port numbers, etc., and populate one or more fields in the data structures. . . .” <i>See Feroz at: 8:6-12:</i> “As discussed more fully below, in one embodiment, flows are identified based on the following flow attributes: 1) source IP address, 2) destination IP address, 3) source port number, 4) destination port number, and 5) protocol (derived from the “protocol” field in IPv4 headers, and the “NextHeader” field in IPv6 headers).” <i>See Feroz at: 8:17-42:</i> “For a TCP flow, for example, packet processor 92 can determine a new data flow by detecting SYN, SYN/ACK, and/or ACK packets. . . . The termination of TCP connections is typically detected by identifying FIN packets; however, the timeout mechanisms discussed above can be used in situations where a FIN packet is not detected.” <i>See Feroz at: 8:61-9:9:</i> “For a TCP/IP packet, packet processor 92 can compute the inside and outside addresses based on the source and destination network addresses of the packet and the direction of the packet flow. Still further, packet processor 92 can also identify which host is the client and which host is the server for a given data flow and store this information in the flow specification or flow object. The identification of a server or client in a given transaction generally depends on the network protocols employed by the hosts. For example, in TCP flows, a client initiates a transaction by transmitting a SYN packet to initiate a TCP connection. Application traffic management device 130 can detect the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	SYN packet and note the source network address of the packet as the client host, and the destination address as the server host. One of ordinary skill in the art will recognize how to identify clients and servers in connection with other networking protocols.” <i>See Feroz at 9:10-36:</i> “In one embodiment, packet processor 92 creates and stores flow objects corresponding to data flows in flow database 135. In one embodiment, flow object attributes include a pointer to a corresponding flow specification object, as well as other flow state parameters, such as TCP connection status, timing of last packets in the inbound and outbound directions, speed information, apparent round trip time, packet count, etc. Flow object attributes further include at least one traffic class identifier (or pointer(s) thereto) associated with the data flow, as well as policy parameters (or pointers thereto) corresponding to the identified traffic class. In one embodiment, flow objects further include a list of traffic classes for which measurement data (maintained by measurement engine 140) associated with the data flow should be logged. In one embodiment, to facilitate association of an existing flow object to subsequent packets associated with a data flow or connection, flow database 135 further maintains a control block hash table including a key comprising a hashed value computed from a string comprising the inside IP address, outside IP address, inside port number, outside port number, and protocol type (e.g., TCP, UDP, etc.) associated with a pointer to the corresponding flow object. According to this embodiment, to identify whether a flow object exists for a given data flow, packet processor 92 hashes the values identified above and scans the hash table for a matching entry. If one exists, packet processor 92 associates the pointer to the corresponding flow object with the packets in the data flow.” <i>"See Feroz at 13:7-38:</i> “As discussed more fully below, flow control module 94 enforces aggregate, partition-based bandwidth utilization controls on data flows traversing access link 21. In one implementation, flow control module 94 monitors bandwidth demand of each partition relative to an aggregate, weighted fair share allocation scheme, and dynamically re-distributes unutilized bandwidth to partitions whose demand for bandwidth exceeds its weighted fair share allocation. As discussed in more detail below, flow control module 94 performs these dynamic allocation adjustments on a continuous basis, as needed. In one implementation, the demand for bandwidth at a given partition can be evaluated in a number of ways. For example, the size of, and rate at which, packets in a flow are received by application traffic management device 130 during a given time interval can be monitored to compute a per-flow rate demand. The per-flow rate demands for all flows falling

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	within a given partition can be aggregated to yield an aggregate rate demand for the partition. In addition, these computed rate values can be smoothed by maintaining a rolling average or weighted moving average across a sliding window of time. Aggregate rate demand for a partition may also be determined by analyzing the number of packets queued in a corresponding partition queue over a given analysis interval. For example, the change in the number of packets queued in a given partition queue over an analysis interval can be analyzed to determine the aggregate rate demand for a given partition. As discussed in more detail below, flow control module 94 can then dynamically allocate unutilized bandwidth to partitions where the aggregate rate demand exceeds their respective bandwidth allocations. In one implementation, the weighting values associated with the partitions determines in part how unutilized bandwidth is distributed.” <i>See Feroz at 13:39-64:</i> “In addition, flow control module 94, in some implementations, can optionally apply per-flow bandwidth utilization controls, on data flows traversing access link 21. In one implementation, flow control module 94 includes a partitioning module operative to enforce aggregate bandwidth utilization controls (e.g., partitions), and a per-flow rate control module operative to apply per-flow rate controls on data flows. In one implementation, flow control module 94 implements the deterministic partition scheduling functionality disclosed in U.S. application Ser. No. 10/108,085, incorporated by reference herein. Furthermore, as discussed more fully below, flow control module 94 includes functionality for dynamically adjusting aggregate bandwidth allocation among partitions. In one implementation, flow control module 94 also includes processes that dynamically adjust per-flow bandwidth allocation within partitions. A partition operates to manage bandwidth for aggregate data flows associated with a traffic class. As discussed above, a partition is essentially a division of the capacity of access link 21. Still further, flow control module 94, in some implementations, may also implement per-flow rate policies on the data flows. In one implementation, flow control module 94 implements the TCP Rate Control technologies disclosed in U.S. Pat. No. 6,038,216 to control the rate at which transmitters send data and therefore the amount of data that is queued in buffers at application traffic management device 130.” <i>See Feroz at 14:13-20:</i> Incorporated by reference U.S. Pat. No. 6,038,216 discloses “the window size is set to allow the specified rate given the current round trip time ("RTT") and the detected speed.” “Similar to the traffic class tree, the root partition node represents the capacity of the access link in either the inbound or outbound directions. In one implementation, separate partition configurations and controls can be maintained for data flows traveling in the inbound or outbound directions

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	relative to network 40. In other implementations, a single partition configuration can be applied to both inbound and outbound data flows.” <i>See Feroz at 14:21-41:</i> “Once a partition configuration is established, in one implementation, a user can simply assign a weight to each partition node, which essentially expresses the priority of the network traffic class(es) falling within the partition relative to other traffic classes. Optionally, a user may also assign a minimum aggregate bandwidth allocation to one or more of the selected partition nodes. As discussed in more detail below, each partition is then allocated a fair share of available bandwidth based on the weights assigned to the partitions, limited by the minimum and maximum bandwidth settings configured for any partitions. In another implementation, a user may also specify a maximum bandwidth allocation for one or more partitions. In one implementation, unless the user specifically configures a weighting value for at least one partition in a given level of the hierarchy, by default, each sibling partition is weighted equally. Additionally, if a user configures a weighting value for only one partition, each sibling partition is configured with a default weighting value. In one implementation, administrator interface 150 requires the user to enter weighting values for all sibling partitions after a user initially configures a weighting value for a selected partition.” <i>See Feroz at 14:42-58:</i> “After the user has completed a configuration, administrator interface 150, in one implementation, processes the configuration to create an aggregate bandwidth allocation scheme that includes partitions, and optionally per-flow rate policies, based on the received configuration. For example, partition queues are created for each leaf partition, and weighted, fair share bandwidth allocations are computed for the partitions, as discussed in more detail below. In addition, in one implementation, each active data flow within a given partition is also guaranteed a fair share of the bandwidth allocated to the partition. The aggregate fair share allocation for each partition is generally stable, unless there are configuration changes. However, the fair share allocation for each flow changes dynamically based on the number of active flows in the partition. In addition, the actual bandwidth allocated to a given partition at a given point in time may change dynamically in response to network traffic demands”

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	See Feroz at 15:6-14: $\text{Aggregate Weight} = \sum \text{Weights from Child Nodes}$ $\text{Allocation Per Partition (c)} = \frac{\text{Parent Allocation} * \text{Configured Weight of Partition (c)}}{\text{Aggregate Weight}}$ See Feroz at 16:56-17:17: “FIG. 6 provides, for didactic purposes, an exemplary hierarchical partition configuration. Root partition node 82 represents the total capacity of access link 21 in either the outbound or inbound direction. In one implementation, separate data structures and processes are maintained for the outbound and inbound directions. Partition nodes 83 represent a first set of partitions allocating the capacity of access link at a first level. As FIG. 6 shows, each parent partition may include child partitions represented as partition nodes 83 and 84. For didactic purposes, assume that access link 21 is a T1 line and, thus, provides maximum available bandwidth of 1.5 Mbps. Partitions P1, P2, and P3 may be configured with weights, as discussed above, to equally divide access link 21 into 0.5 Mbps partitions at a first level. In addition, child partitions P4 and P5 may further divide parent partition P1 into a 0.2 Mbps partition (P4) and a 0.3 Mbps partition (P5). Similarly, child partitions P6, P7 and P8 may divide parent partition P2 into two 0.1 Mbps partitions (P6 and P8), and a 0.3 Mbps partition (P7). As discussed above, the initial aggregate bandwidth allocations may also be derived from weights associated with the partition nodes. Of course, any suitable partition configuration may be employed, including the configurations illustrated in FIGS. 9A thru 9F, See above. Still further, other configurations are possible. For example, the root node may represent the access link in both the inbound and outbound direction, while child partitions from the root can correspond to traffic encountered at a given network interface (such as interface 71) and destined for egress from a second network interface (such as interface 72). . . .” See Feroz at 18:28-62: “For example, each partition object, in one implementation, may include the following statistics: 1)

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	a flow count (the number of currently active flows); 2) current rate (a weighted moving average in bps of the rate observed for the flows in the class); 3) aggregate compensation demand; and 4) aggregate unutilized fair share. In addition, per-flow statistics are also maintained, such as the current rate of the flow (a weighted moving average over an analysis interval of the rate at which packets are transmitted from application traffic management device), packet count, and the like. One skilled in the art will recognize that the order of processing operations illustrated in FIG. 3 is one possible process flow and that the flow control processes described herein can be performed at other points in the packet processing path.” <i>See Feroz at 19:12-26:</i> “For example, each partition object, in one implementation, may include the following statistics: 1) a flow count (the number of currently active flows); 2) current rate (a weighted moving average in bps of the rate observed for the flows in the class); 3) aggregate compensation demand; and 4) aggregate unutilized fair share. In addition, per-flow statistics are also maintained, such as the current rate of the flow (a weighted moving average over an analysis interval of the rate at which packets are transmitted from application traffic management device), packet count, and the like. One skilled in the art will recognize that the order of processing operations illustrated in FIG. 3 is one possible process flow and that the flow control processes described herein can be performed at other points in the packet processing path.” <i>See Feroz at 19:28-20:4:</i> “FIG. 4A illustrates a process, according to one implementation of the invention, directed to reading packets onto flow and partition queues. . . . In addition, since the packet represents the addition of a new flow to a partition, flow control module 94 re-computes the per-flow bandwidth allocation for the partition (316), as discussed in a following section. . . . In one implementation, if the initial flow rate of the flow is less than the per-flow fair share for the partition (318), flow control module 94 pushes the flow queue onto the high priority queue (322), facilitating the chance for the rate of the flow to increase. Otherwise, flow control module 94 pushes the flow queue onto the low priority queue 60 a corresponding to the identified partition (320). In one implementation, flow control module 94 applies a process directed to the per-flow rate enforcement. In one implementation, the per-flow target rate is enforced by mapping the target rate to an advertised TCP window size, as disclosed in U.S. Pat. No. 6,038,216, and modifying the advertised window size in the TCP ACK fields of packets transmitted back to the transmitter to modify the rate at which subsequent packets are sent. Of course, the per-flow rate control processes may be applied to the packets at any point prior to egress from application traffic management device 130. In addition, the

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	use of TCP Rate Control technologies to control the rate of data transmission from end hosts is an optional feature.” <i>See Feroz at 20:6-20:</i> “FIG. 4B sets forth a method, according to one implementation of the invention, directed to computing a per-flow fair share allocation and a target rate for flows within a given partition. As FIG. 4B illustrates, the per-flow fair share bandwidth allocation is the current aggregate bandwidth allocated to the partition divided by the number of active flows (330). As discussed above, the current number of active flows is maintained in a traffic class or partition object. The per-flow fair share rate can be used in a variety of ways. In one implementation, the per-flow fair share rate can be used as the basis of determining whether packets corresponding to a flow are placed on the high or low priority partition queue. In another implementation, the per-flow fair share rate can be used in a process that computes a target rate for each flow within a partition.” <i>See Feroz at 20:21-45:</i> “According to one implementation, the allocation of bandwidth to flows within a given partition is an optional step. In such an implementation, as FIG. 4B illustrates, flow control module 94 computes the aggregate compensation demand and the aggregate bandwidth underutilization for the partition (331). Compensation demand occurs when the current rate of a flow exceeds a threshold based on a percentage of the target rate of the flow. In one implementation, the compensation demand threshold is 94% of the per-flow target rate; however, this percentage can vary based on various engineering and design considerations. The compensation demand for a flow, in one implementation, is a fixed percentage of the detected current rate for the flow. As discussed above, the current rate can be estimated, in one implementation, by monitoring the size of incoming packets and the time at which they are received. Aggregate compensation demand for a partition is the sum of the compensation demands for each flow. Oppositely, bandwidth underutilization (unutilized fair share) occurs when the current rate associated with the flow is less than a percentage of the target rate, and is computed based on the difference between the per-flow fair share bandwidth allocation and the current rate. In one implementation, the aggregate bandwidth underutilization for a partition is thus the sum of the individual bandwidth underutilization values for the flows in the partition.” <i>See Feroz at claim 1.</i>
[11d] transmitting, by the	<i>See claim element [1c] supra.</i>

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
network controller, the over-subscription ratio to the network switching element based on the over-subscription ratio and a threshold value;	
[11e] determining, by the network controller, a fair-share bandwidth allocation for the target port based on the over-subscription ratio and the threshold value, the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and	<i>See claim element [1c] supra.</i>
[11f] adjusting, by the network controller, the bandwidth allocation for the target port based on the fair-share bandwidth allocation.	<i>See claim element [1d] supra.</i>
[17pre] An edge switch for adjusting bandwidth allocation in a communications network, the edge switch including a target port, the edge switch configured to:	To the extent the preamble is limiting, Feroz discloses this preamble. <i>See claim element [1pre] supra.</i>
[17a] monitor a data flow traversing the target port;	<i>See claim element [1a] supra.</i>
[17b] determine a bandwidth allocation for the target port, the bandwidth allocation for the target port being a bandwidth that is currently	<i>See claim element [1b] supra.</i>

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
allocated for the data flow;	
[17c] determine a fair-share bandwidth allocation for the target port, the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and	<i>See claim element [1c] supra.</i>
[17d] adjust the bandwidth allocation for the target port based on the fair-share bandwidth allocation.	<i>See claim element [1d] supra.</i>
[20pre] A Software Defined Networking (SDN) controller for adjusting bandwidth allocation in a communications network, the SDN controller being configured to control an aggregation switch, the aggregation switch including a target port, the SDN controller is configured to:	To the extent the preamble is limiting, Feroz discloses this preamble. <i>See Feroz at Abstract:</i> “Methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that prevents spiraling decline of data flow rates.” <i>See Feroz at 4:46-61:</i> “The present invention provides methods, apparatuses and systems directed to an aggregate bandwidth utilization control scheme including fair share bandwidth allocation and dynamic allocation of bandwidth in response to detected traffic utilization. In one implementation, the present invention includes a weighted, fair share aggregate bandwidth allocation mechanism that dynamically responds to observed bandwidth utilization to provide unutilized or excess bandwidth to flows and partitions that require it. In another implementation, the present invention features a weighted fair share allocation scheme for hierarchical partition configurations. In other implementations, the present invention provides a per-flow target rate assignment mechanism that

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	prevents spiraling decline of data flow rates. These and other aspects of various implementations of the present invention will become apparent from the drawings and description provided below.” <i>See Feroz at 6:56-17:17:</i> “FIG. 6 provides, for didactic purposes, an exemplary hierarchical partition configuration. Root partition node 82 represents the total capacity of access link 21 in either the outbound or inbound direction. In one implementation, separate data structures and processes are maintained for the outbound and inbound directions. Partition nodes 83 represent a first set of partitions allocating the capacity of access link at a first level. As FIG. 6 shows, each parent partition may include child partitions represented as partition nodes 83 and 84. For didactic purposes, assume that access link 21 is a T1 line and, thus, provides maximum available bandwidth of 1.5 Mbps. Partitions P1, P2, and P3 may be configured with weights, as discussed above, to equally divide access link 21 into 0.5 Mbps partitions at a first level. In addition, child partitions P4 and P5 may further divide parent partition P1 into a 0.2 Mbps partition (P4) and a 0.3 Mbps partition (P5). Similarly, child partitions P6, P7 and P8 may divide parent partition P2 into two 0.1 Mbps partitions (P6 and P8), and a 0.3 Mbps partition (P7). As discussed above, the initial aggregate bandwidth allocations may also be derived from weights associated with the partition nodes. Of course, any suitable partition configuration may be employed, including the configurations illustrated in FIGS. 9A thru 9F, See above. Still further, other configurations are possible. For example, the root node may represent the access link in both the inbound and outbound direction, while child partitions from the root can correspond to traffic encountered at a given network interface (such as interface 71) and destined for egress from a second network interface (such as interface 72). . . .” <i>See Feroz at 18:28-62:</i> “FIG. 3 illustrates, packet processor 92 updates, for existing data flows, attributes of the flow object in response to the packet such as the packet count, last packet time, and the like (208). Packet processor 92 can also perform other operations, such as analyzing the packets for connection state information. For example, packet processor 92 can inspect various TCP flags to determine whether the received packet is part of a new data flow or represents a change to an existing data flow (such as the first data packet after the TCP handshake). Methods for determining new data flows and assigning packets to existing data flows are well known in the art and also depend on the particular transport layer protocol employed. For a TCP packet, packet processor 92 can determine a new data flow by detecting SYN and/or SYN/ACK packets. However, a new data flow can simply be a data flow for which there is no corresponding flow object in flow database 135. . . . In one implementation, this separate process can also be passed pointers to TCP FIN packets to adjust

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
	various data structures, such as reallocating flow object space to a pool of available memory, adjusting flow counts in traffic class or partition objects.” <i>See Feroz at 20:6-20:</i> “FIG. 4B sets forth a method, according to one implementation of the invention, directed to computing a per-flow fair share allocation and a target rate for flows within a given partition. As FIG. 4B illustrates, the per-flow fair share bandwidth allocation is the current aggregate bandwidth allocated to the partition divided by the number of active flows (330). As discussed above, the current number of active flows is maintained in a traffic class or partition object. The per-flow fair share rate can be used in a variety of ways. In one implementation, the per-flow fair share rate can be used as the basis of determining whether packets corresponding to a flow are placed on the high or low priority partition queue. In another implementation, the per-flow fair share rate can be used in a process that computes a target rate for each flow within a partition.”
[20a] receive data flow information from the network switching element, the data flow information including information about data flows traversing of the target port;	<i>See claim element [1a] supra; see claim element [11a] supra.</i>
[20b] determine a bandwidth allocation of the target port based on the data flow information, the bandwidth allocation of the target port being a bandwidth that is currently allocated for each of the data flows traversing the target port;	<i>See claim element [1b] supra; see claim element [11b] supra.</i>
[20c] determine an over-subscription ratio, the over-subscription ratio being a ratio of the bandwidth allocation of	<i>See claim element [11c] supra.</i>

Exhibit E13 – Invalidity of U.S. Patent No. 9,450,884 in view of Feroz – US Patent No. 7,453,804

Claim	Disclosure
the target port to a number of data flows traversing the target port; and	
[20d] transmit the over-subscription ratio to the aggregation switch based on the over-subscription ratio and a threshold value;	<i>See claim element [1c] supra; see claim element [11d] supra.</i>
[20e] determine a fair-share bandwidth allocation for the target port based on the over-subscription ratio and the threshold value, the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and	<i>See claim element [1c] supra; see claim element [11e] supra.</i>
[20f] adjust the bandwidth allocation for the target port based on the fair-share bandwidth allocation.	<i>See claim element [1d] supra; see claim element [11f] supra.</i>