

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS INC.,
Petitioner

IPR2025-00429
U.S. Patent No. 9,450,884

**DECLARATION OF PAUL MIN, PH.D.,
UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION
FOR *INTER PARTES* REVIEW**

TABLE OF CONTENTS

I.	INTRODUCTION	3
II.	QUALIFICATIONS AND PROFESSIONAL EXPERIENCE	5
III.	LEVEL OF ORDINARY SKILL IN THE ART	7
IV.	RELEVANT LEGAL STANDARDS	8
V.	OVERVIEW OF THE '884 PATENT	9
VI.	CLAIM CONSTRUCTION	14
VII.	HOW THE CLAIMS ARE UNPATENTABLE	15
VIII.	FEROZ AND RIDDLE RENDER OBVIOUS CLAIMS 1 AND 17	15
	A. Summary of Feroz	15
	B. Summary of Riddle	19
	C. Reasons to Combine Feroz and Riddle	21
	D. Detailed Analysis of Claims.....	24
	1. Claim 1	24
	2. Claim 17	75
IX.	PETERSEN, ALONE OR IN COMBINATION WITH PALACHARLA, RENDERS OBVIOUS CLAIMS 1 AND 17	78
	A. Summary of Petersen	78
	B. Summary of Paracharla	80
	C. Reasons to Combine Petersen and Paracharla	80
	D. Detailed Analysis of Claims.....	83
	1. Claim 1	83
	2. Claim 17	103
X.	CONCLUSION.....	105

I, Paul Min, Ph.D., do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Cisco Systems, Inc. in the matter of the *Inter Partes* Review of U.S. Patent No. 9,450,884 (“the ’884 patent”) to Hwang et al.

2. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1 and 17 (“the Challenged Claims”) of the ’884 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

- a. the ’884 patent, Ex.1001;
- b. the prosecution history of the ’884 patent (“’884 File History”), Ex.1002;
- c. U.S. Patent No. 7,453,804 to Feroz et al. (“Feroz”), Ex.1005;
- d. U.S. Patent No. 7,406,522 to Riddle (“Riddle”), Ex.1006;

e. U.S. Patent Publication No. 2007/0248009 to Petersen (“Petersen”), Ex.1008; and

f. U.S. Patent Publication No. 2009/0097495 to Palacharla (“Palacharla”), Ex.1009.

5. In forming the opinions expressed below, I have considered: the documents listed above; the relevant legal standards, including the standard for obviousness; and my own knowledge and experience based upon my work in the field of network communications as described below, as well as portions of the following additional materials:

a. Robert Breyer and Sean Riley, SWITCHED AND FAST ETHERNET, 2d ed. (Ziff-Davis Press 1996), Ex.1010;

b. The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specifications, version 2.0 (Nov. 1982) (“DIX Ethernet”), Ex.1014;

c. US 2003/0174649 to Shankar et al., Ex.1018;

d. US 7,869,366 to Muppala et al., Ex.1019;

e. US 2013/0100858 to Kamath et al., Ex.2021.

6. Unless otherwise noted, all **emphasis** in any quoted material has been added. Claim terms are *italicized*.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

7. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which can be found in Exhibit 1004. The following is a brief summary of my relevant qualifications and professional experience.

8. I have extensive experience in electrical engineering, computer technology, and communications networks in both academic and professional settings. I am a tenured professor in the Department of Electrical and Systems Engineering at Washington University in Saint Louis. I have served as a professor at Washington University since 1990 when I began as an Assistant Professor in the Department of Electrical Engineering. In 1996, I became a tenured professor and served as an Associate Professor for eighteen years. From 2000 to 2002, I served as the Chair of Graduate Curriculum in the Department of Electrical and Systems Engineering. From 2011 to 2014, I served as the Chair of Undergraduate Curriculum of the Department. In 2015, I was promoted to Senior Professor and have been serving in that role since.

9. My teaching at Washington University centers on including computer communications and systems analysis, data networks, queuing systems and discrete stochastic processes, signaling and control of communications networks, reliability

and quality control, communication theory, and transmission system and multiplexing, just to name a few.

10. In 1987, I received a Ph.D. in Electrical Engineering from the University of Michigan. I received a Master's Degree in Electrical Engineering in 1984 and a Bachelor's Degree in Electrical Engineering in 1982, both from the University of Michigan.

11. I have also founded and managed multiple companies in the field of communications technology. In 1999, I founded Erlang Technology, Inc. in St. Louis, Missouri. Erlang provided an award-winning product, [...], employed over 70 employees, and raised \$40M in total capital. In 1997, I founded and managed MinMax Technologies, Inc., which designed and produced fabless semiconductors and high performance switching ASICs.

12. As outlined in detail in my attached curriculum vitae, I have served on multiple technical program committees, received multiple awards for research, academic papers, competitions, and other various achievements in my field of study. I am also the inventor or coinventor of eleven issued patents myself, issued from 1995 to 2020 in the field of electrical engineering, including network devices, packet switching, etc. I have published nearly 100 academic papers covering topics ranging from intrusion detection for computer networks, performance evaluation,

infrastructure arrangement, fair queuing, network architecture and design, real-time adaptive bandwidth allocation, and reliability analysis.

13. I have served as an expert witness and technical consultant in over 75 litigation and *inter partes* review matters concerning computer networks, wireless networks at both the protocol and hardware levels, and telecommunications. I am extremely familiar with the technologies involved with the '884 patent, including network design and methods of bandwidth allocation within network devices.

III. LEVEL OF ORDINARY SKILL IN THE ART

14. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

15. A POSITA in the field of the '884 patent, as of its priority date of June 11, 2014, would have been someone knowledgeable and familiar with network design and methods of bandwidth allocation within network devices. Such a POSITA would have a master's degree in computer science, computer engineering, electrical engineering, or equivalent training, and approximately two years of experience working in the field of packet-based computer networking. Additional work experience can substitute for specific educational background, and vice versa.

16. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my own experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the priority date of the '884 patent.

IV. RELEVANT LEGAL STANDARDS

17. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the '884 patent, I am relying on certain basic legal principles that Cisco's counsel has explained to me.

18. I understand that prior art to the '884 patent includes patents and printed publications in the relevant art that predate the priority date of the '884 patent. For purposes of this Declaration, I am applying June 11, 2014, as the priority date of the '884 patent.

19. I have been informed by Cisco's counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by Cisco's counsel that the obviousness analysis considers factual inquiries, including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

20. I have been further informed by Cisco’s counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness. These rationales include: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

21. Also, I have been informed and understand that obviousness does not require physical combination/bodily incorporation, but rather consideration of what the combined teachings would have suggested to a POSITA at the time of the alleged invention.

V. OVERVIEW OF THE ’884 PATENT

22. The ’884 patent relates generally to “[m]ethods and systems of adjusting bandwidth allocation.” ’884 patent at Abstract. The ’884 patent describes “a network element” for “monitoring a data flow traversing a target port.” ’884 Patent, Abs. The network element “determin[es] a bandwidth allocation for the

target port, determin[es] a fair-share bandwidth allocation for the target port, and adjust[s] the bandwidth allocation for the target port based on the fair-share bandwidth allocation.” ’884 Patent, Abs.

23. The ’884 patent explains that various network devices may implement the method, including “an edge switch, aggregation switch, or core switch as shown in FIG. 1,” shown below. ’884 Patent, 7:39-44.

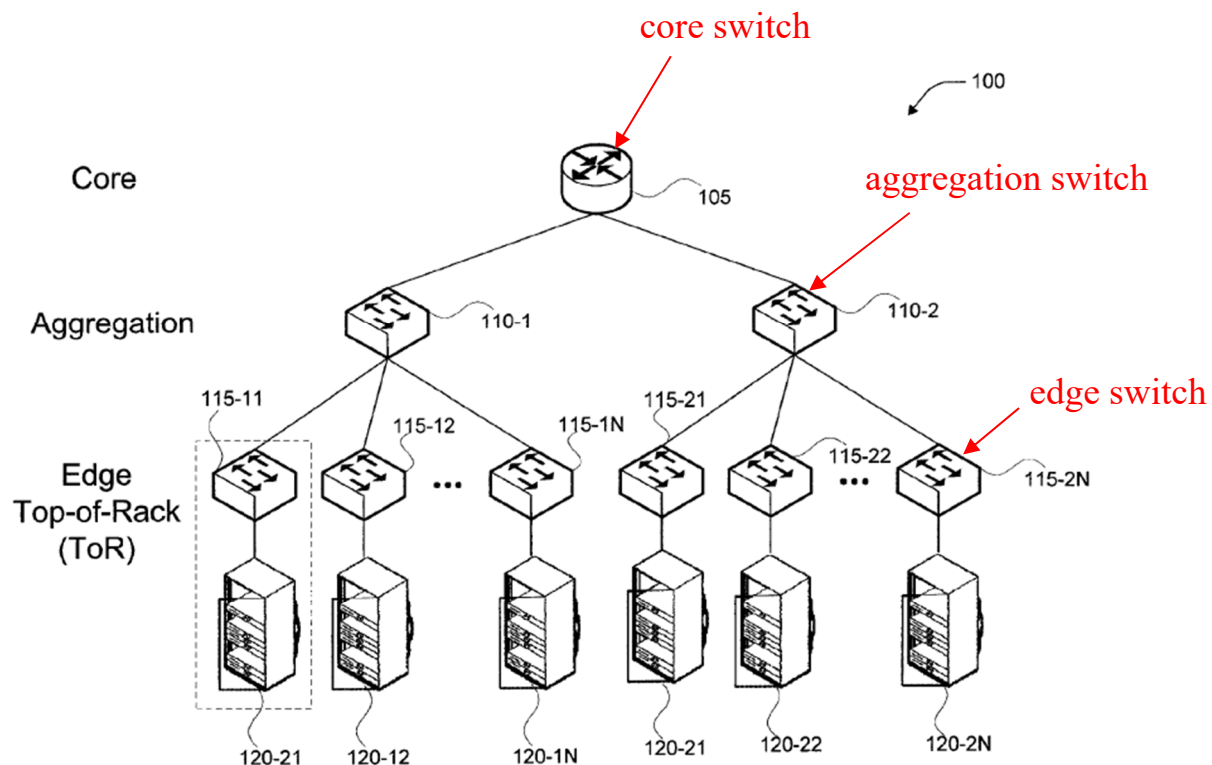


Fig.1

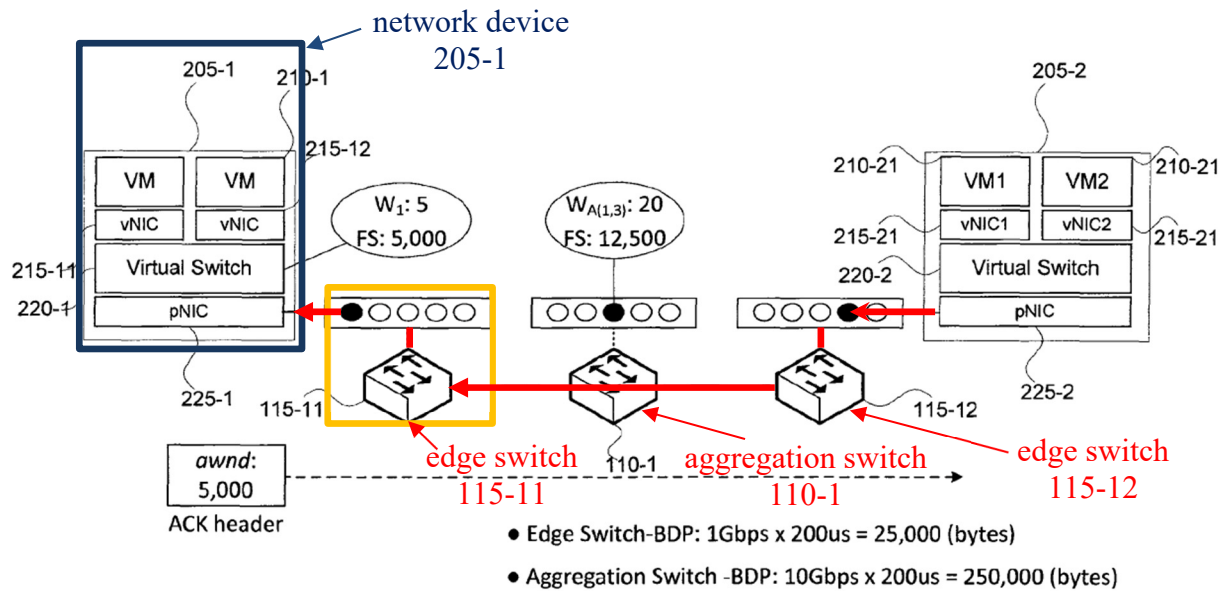
’884 patent at Fig. 1 (annotated)

24. The ’884 patent explains that a switch which implements its method may “include multiple connection interfaces...including Ethernet, Fibre Channel,

ATM, ITU-T, G.hn, 802.11, and/or other like connection interfaces.” ’884 Patent, 8:15-19. As illustrated in Figure 1, above, the edge switches 115 are connected to “[s]erver stacks 120.” ’884 Patent, 8:58-60. The server stacks 120 “provide services for users connected to a network.” ’884 Patent, 8:58-60.

25. In more detail, the ’884 patent describes an “SDN controller 230” “at each edge switch 115” which “monitors all data flows that traverse each port.” ’884 Patent, 12:50-52. “Additionally, in such embodiments the SDN controller 230 computes, calculates, or otherwise determines a fair-share bandwidth allocation and/or weighted fair-share bandwidth allocation for each data flow.” ’884 Patent, 12:61-64. “The fair-share bandwidth allocation is a proportional allocation of a total bandwidth of the network element.” See, e.g., ’884 Patent, Abs., 1:57-58, 3:12-13, 4:9-13. “The fair-share bandwidth allocation is a proportional allocation of a total bandwidth, such that each port of an edge switch is allocated a proportional amount of bandwidth. The weighted fair-share bandwidth allocation is a proportional allocation of a total bandwidth, such that each port of an edge switch is allocated a proportional amount of bandwidth based on a weight value associated with data flows traversing each port.” ’884 Patent, 12:65-13:5.

26. Figure 4 of the ’884 patent illustrates a data flow that traverses port 4 of edge switch 115-12, port 3 of aggregation switch 110-1, and port 1 of edge switch 115-11. ’884 Patent, 14:51-53.



'884 patent at Fig. 4 (annotated).

27. The SDN controller 230 determines a fair-share bandwidth allocation for data flows that traverse a port (e.g. port 1) of an edge switch (e.g. edge switch 115-11). In particular, when a packet arrives at network device 205-1, the SDN controller determines the number of data flows that traverse the target port (e.g. port 1) of edge switch 115-11. '884 Patent, 14:56-59.

28. The SDN controller then determines the uplink capacity (or "link capacity") of the target port of edge switch 115-11 and a RTT of the data flows traversing the target port (e.g. port 1), to calculate a bandwidth delay product (BDP) according to Equation 1: $\text{BDP} = \text{link_capacity}_i \times \text{RTT}$, where link_capacity_i is the

link capacity of a target port (e.g. port 1), and RTT is the round trip time of at least one data flow traversing the target port (e.g. port 1). '884 Patent, 14:59-15:1.

29. In one example, when a packet arrives at network device 205-1, the SDN controller determines that port 1 (i.e. the target port) of edge switch 115-11 is shared by five (5) data flows by counting a number of data flows traversing port 1. '884 Patent, 14:59-15:1. Assuming that the link capacity of the target port is 1Gbps, and assuming a RTT of the data flows traversing port 1 of edge switch 115-11 is 200 μ s, the BDP for port 1 is 25,000 bytes. '884 Patent, 15:10-13.

30. The SDN controller then determines a weighted fair-share bandwidth allocation for data flows that traverse the port 1 of edge switch 115-11 according to Equation 2: $\text{weighted_fair_share}_{ki} = \text{BDP} \times (w_k/W_i)$, where $\text{weighted_fair_share}$ is the weighted fair-share bandwidth allocation of a data flow k traversing a target port of a physical network interface card (pNIC), w_k is the weight value assigned to or otherwise associated with data flow k , and W_i is the total weight value of the i th pNIC. '884 Patent, 15:14-29. "It should be noted that, in the instance that each of the data flows are treated equally (i.e., where each w_k has an equal value, such as 1), then W_i will indicate a number of active data flows traversing a virtual switch [of network device 205-1] through the i th pNIC." '884 Patent, 15:34-38.

31. "Referring back to FIG. 4, the SDN controller 230 determines that the weight value assigned to or otherwise associated with the data flow k is 5 (W_1 as

shown in FIG. 4). Using equation 2, and because the weights for each data flow traversing port 1 are equal, the weighted fair-share bandwidth allocation for port 1 of edge switch 115-11 is 5,000 bytes.” ’884 Patent, 15:39-45.

32. As I will explain below, allocating bandwidth in accordance with fair-share bandwidth allocations was already a known technique as of the priority date of the ’884 patent.

VI. CLAIM CONSTRUCTION

33. It is my understanding that in order to properly evaluate the ’884 patent, the terms of the claims must first be interpreted. It is my understanding that for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry.

34. I have reviewed the entirety of the ’884 patent, as well as its prosecution history. It is my opinion that for purposes of applying the prior art presented herein to evaluate the patentability of the Challenged Claims no claim terms require construction. Accordingly, I have analyzed the claims consistent with their ordinary and customary meaning as would be understood by a POSITA.

VII. HOW THE CLAIMS ARE UNPATENTABLE

35. The discussion in this Declaration provides a detailed analysis of how the asserted prior art references teach each limitation of the Challenged Claims.

36. As part of my analysis, I have considered, and discuss in detail, the scope and content of the prior art and any differences between the alleged invention and the prior art.

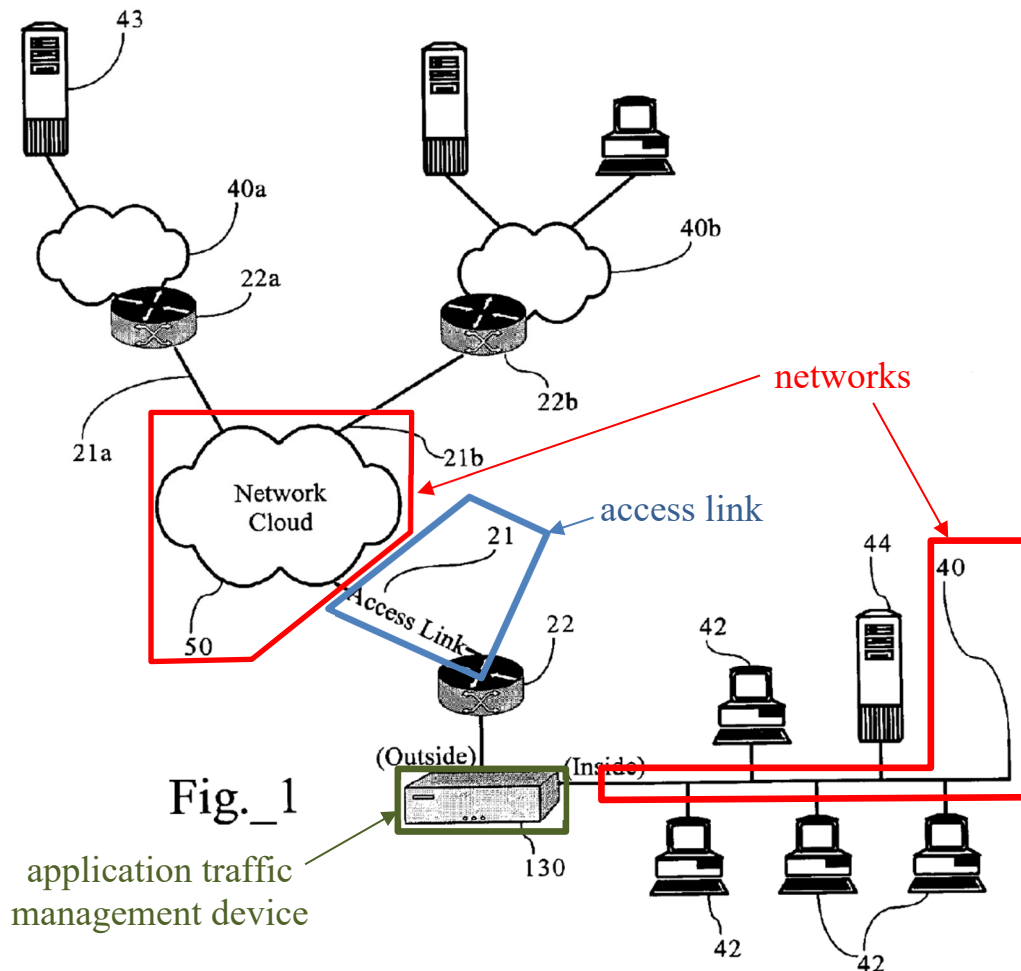
37. It is my opinion that the alleged invention recited in the Challenged Claims would have been obvious in view of the teachings of the asserted prior art and the knowledge of a POSITA.

VIII. FEROZ AND RIDDLE RENDER OBVIOUS CLAIMS 1 AND 17

A. Summary of Feroz

38. Like the '884 patent, Feroz discloses methods of adjusting bandwidth allocation to optimize bandwidth. In particular, Feroz describes “[m]ethods, 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.” Feroz at Abstract. Feroz further describes calculating “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.” Feroz at Abstract.

39. To implement this method, Feroz describes a network with an “application traffic management device 130” located at the edge of a network 40 as shown in Figure 1.



Ex.1005, Fig. 1 (annotated).

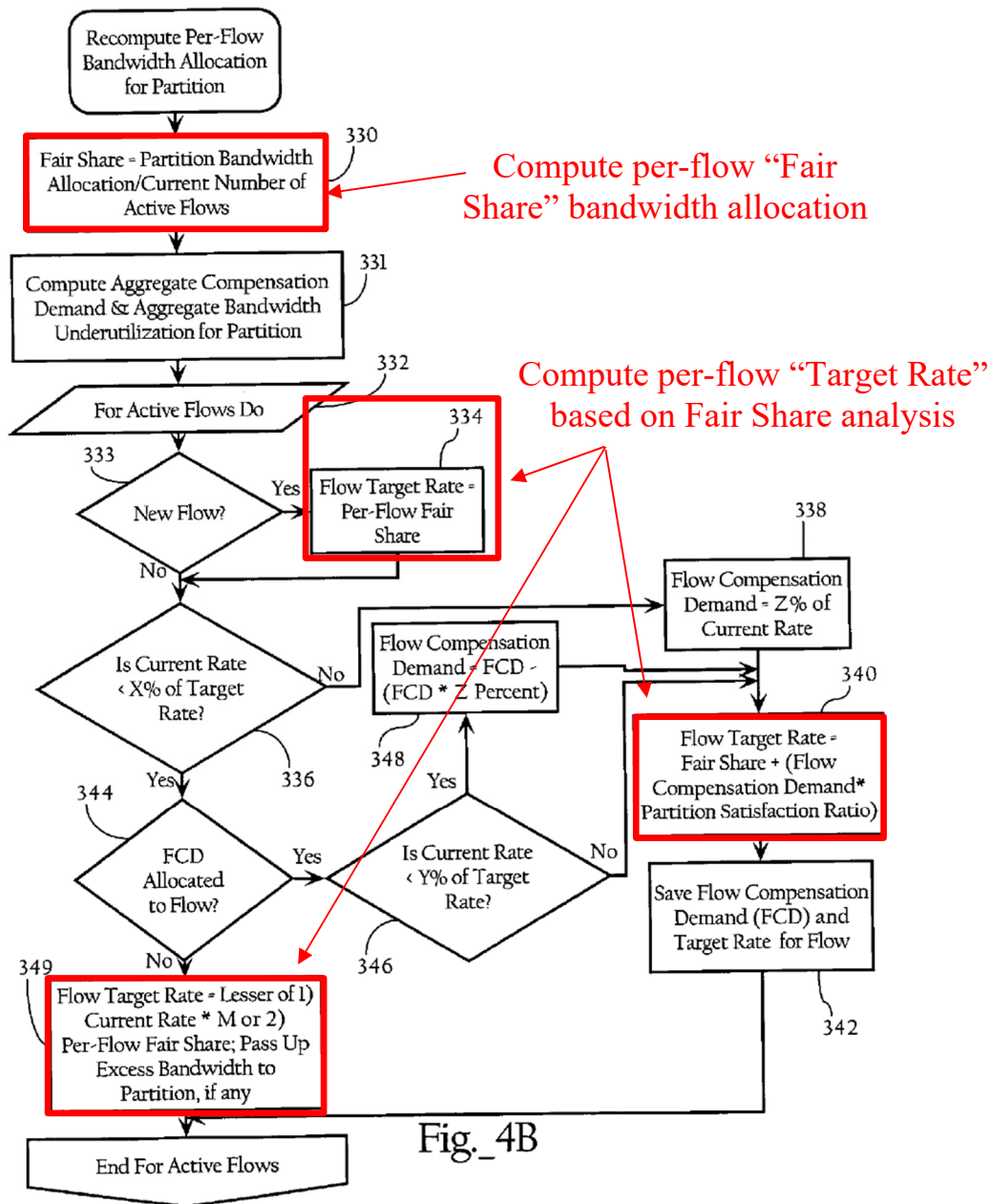
40. The “application traffic management device 130 is operative to classify and manage data flows traversing access link 21” using a “flow control module 94.” Feroz at 5:64-67, 7:5-9, Fig. 2. A POSITA would have understood that Feroz’s application traffic management device 130 is able to do this, even though it is not

itself directly connected to access link 21, because both access link 21 and the application traffic management device 130 are on the only disclosed path between network 40 and network 50. Thus, all (or substantially all) traffic that traverses access link 21 also traverses the application traffic management device 130. By controlling the packet traffic flowing through itself, the application traffic management device 130 controls the packet traffic flowing over the access link 21.

41. Feroz teaches that application traffic management devices use aggregate bandwidth policies to allocate available bandwidth among “partitions.” For example, available bandwidth may be partitioned based on policies where certain partitions have preference and thus are allocated a higher allocation of bandwidth relative to partitions that have lower allocations of bandwidth. Feroz at 3:55-4:16.

42. Within a partition there may be a number of “data flows.” Feroz describes that “per-flow” rate policies can allocate bandwidth to particular “data flows” falling within a partition by setting a “per-flow” target rate. Feroz at 4:17-36.

43. The application traffic management device 130 calculates a “per-flow fair share allocation and a target rate for flows within a given partition,” as shown in Figure 4B, below. Feroz at 20:6-10.



Feroz at Fig. 4B.

44. Feroz describes that its bandwidth allocation process is “dynamic.” Feroz at 4:46-50. Specifically, after a flow target rate is initially computed for a data flow, the management device 130 continues to monitor bandwidth utilization and

network performance and can “dynamically adjust bandwidth allocation across flows and partitions to improve or manage network performance” on a “continuous basis, as needed.” Feroz at 7:24-28, 13:14-17.

B. Summary of Riddle

45. Riddle also generally relates to the field of packet-based computer networking. For example, Riddle discusses using the well-known “TCP/IP protocol suite” and describes network communications in the context of the “Open Systems Interconnection (OSI) model,” both of which a POSITA would have been well-acquainted with. Riddle at 2:8-13. And like Feroz and the ’884 Patent, Riddle’s specification is directed toward addressing the issue of controlling bandwidth allocations. *See, e.g.*, Riddle at 5:36-39, 6:35-41, 17:11-22.

46. Also like Feroz, Riddle describes making dynamically allocating bandwidth among various network traffic “partitions.” Riddle describes these partitions being arranged in a hierarchy:

Partitions are arranged in a hierarchy— that is, partitions can contain partitions. For example, a static parent partition can be divided into multiple static child partitions. In one embodiment, at the highest level, a partition exists for all available outbound bandwidth, while another partition exists for all available inbound bandwidth across the particular access link. These partitions are then sub-dividable to form a hierarchical

tree. For example, an enterprise employing static partitions may define a static partition for a PeopleSoft software application traffic class, and sub-divide this parent partition into a large burstable child partition for its human resources department and a smaller burstable child partition for the accounting department.

Riddle at 6:55-67.

47. The highest level of partition, which Riddle calls the root node, corresponds to the network interface itself, thus encompassing all traffic flowing in either direction through the network interface:

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).

Riddle at 17:12-17.

48. Alternatively, Riddle describes there being multiple “highest level” partitions, with separate partitions for inbound and outbound traffic:

In one embodiment, at the highest level, **a partition exists for all available outbound bandwidth**, while **another partition exists for all available inbound bandwidth** across the particular access link. These partitions are then sub-dividable to form a hierarchical tree.

Riddle at 6:58-61.

C. Reasons to Combine Feroz and Riddle

49. A POSITA would have found it obvious to combine features from Feroz and Riddle because Feroz expressly refers to, and incorporates by reference, the entire disclosure of Riddle:

This application makes reference to the following commonly owned U.S. patent applications and patents, which are incorporated herein by reference in their entirety for all purposes:...

U.S. patent application Ser. No. 09/966,538, in the name of Guy Riddle, entitled “Dynamic Partitioning of Network Resources;”

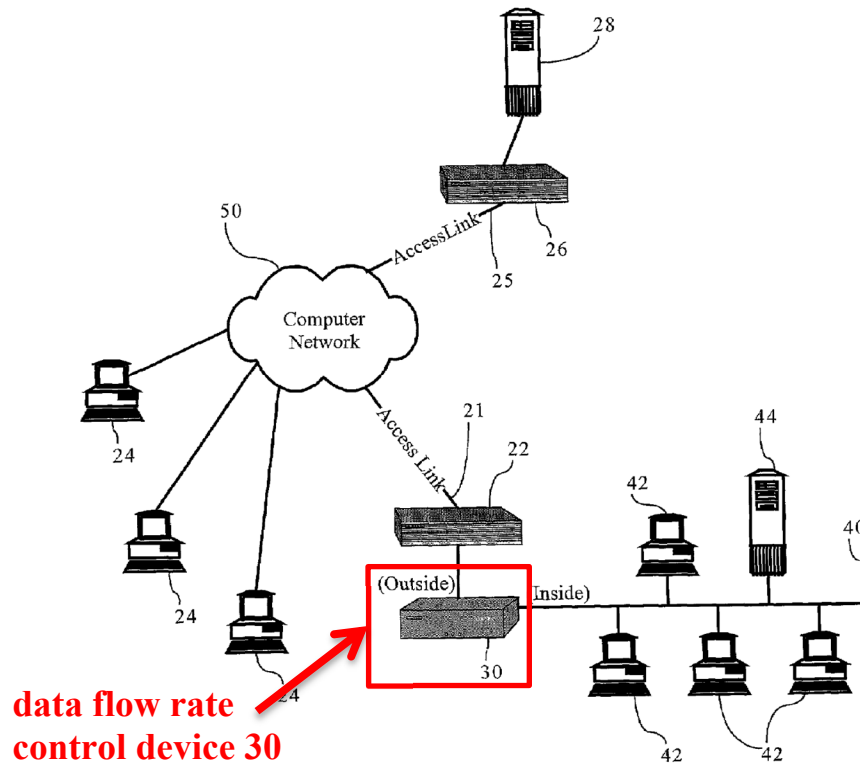
Feroz at 1:7-10 and 1:56-58.

50. The serial number 09/966,538 corresponds to Riddle:

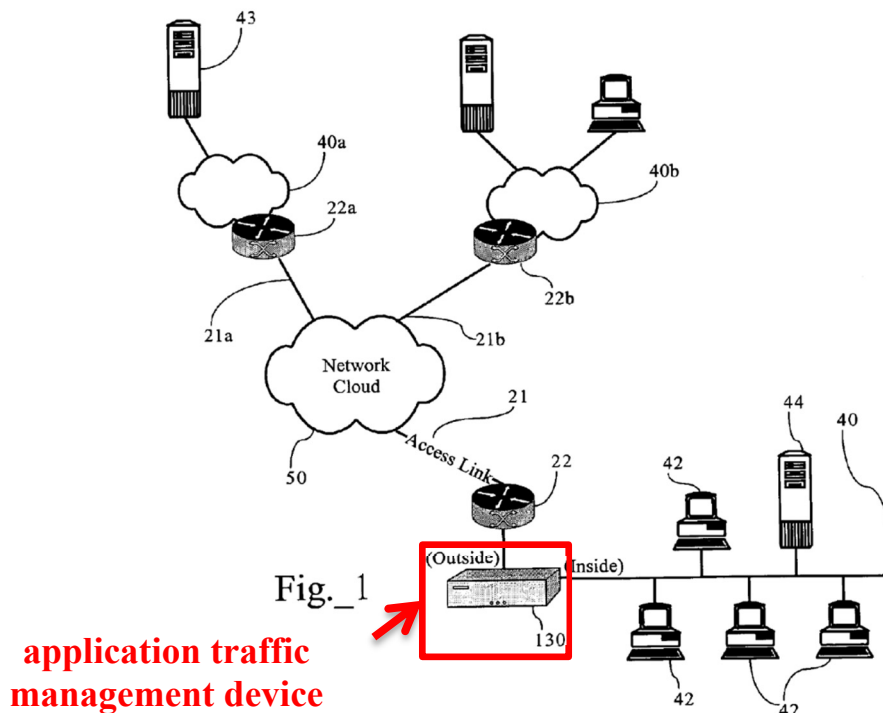
(12) United States Patent		(10) Patent No.:	US 7,406,522 B2	
Riddle		(45) Date of Patent:	Jul. 29, 2008	
(54) DYNAMIC PARTITIONING OF NETWORK RESOURCES		6,546,415 B1 *	4/2003	Park 709/202
		6,771,661 B1 *	8/2004	Chawla et al. 370/468
(75) Inventor: Guy Riddle , Los Gatos, CA (US)		6,934,745 B2	8/2005	Krautkremer
		6,952,735 B1 *	10/2005	Aune 709/228
(73) Assignee: Packeteer, Inc. , Cupertino, CA (US)		6,968,323 B1 *	11/2005	Bansal et al. 705/80
		2002/0194326 A1 *	12/2002	Gold et al. 709/224
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 841 days.				
(21) Appl. No.: 09/966,538		OTHER PUBLICATIONS		
		Packeteer, “PacketShaper Reference Guide”, versions 3.1-5.1.*		

51. Thus, the express statement in Feroz incorporating the ideas and concepts from Riddle would have prompted a POSITA to consider them together.

52. While Feroz alone renders obvious many of the claim limitations, as I discuss below, I note that Riddle provides supplemental descriptions of many concepts and components described in Feroz. For example, Feroz describes an application traffic management device 130 that is “operative to classify and manage data flows traversing access link 21.” Feroz at 5:65-67. The application traffic management device 130 is on a data path connecting a network 40 to a network 50. Feroz, Fig. 1. Riddle provides a similar description of a “data flow rate control device 30” that is “operative to classify data flows and... allocate bandwidth utilization across access link 21.” Riddle, 4:64-67. The data flow rate control device 30 is similarly located between a network 40 and a network 50. From the similar descriptions and figures in Feroz and Riddle, it would have been obvious to a POSITA to apply features from Riddle’s data flow rate control device 30 to Feroz’s application traffic management device 130.



Riddle at Fig. 1 (annotated).



Feroz at Fig. 1 (annotated).

D. Detailed Analysis of Claims

1. Claim 1

- a. **[1.0] *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:***

53. To the extent that the preamble is limiting, Feroz renders it obvious.

54. First, Feroz teaches “methods” that are directed to “bandwidth allocation” and “dynamic allocation of bandwidth” in a network:

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.

Feroz at Abstract.

55. Feroz's techniques for the dynamic allocation of bandwidth render obvious a "*method of adjusting bandwidth allocation.*"

"...by a network switching element in a communications network..."

56. **Second**, Feroz's bandwidth allocation techniques are employed by an application traffic management device 130 that classifies and manages data flows and dynamically "adjust[s] bandwidth allocations" across flows and partitions to improve or manage network application performance:

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.

Feroz at 5:62-6:3.

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.**

Feroz at 7:10-28.

57. Feroz illustrates in Fig. 1 that the application traffic management device 130 is at the edge of network 40. Feroz describes that network 40 can be a local area network, a wide area network, or any other suitable network, and that the application traffic management device 130 classifies and manages data flows:

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 40a & 40b, **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.

Feroz at 5:49-6:3.

58. Thus, network 40 corresponds to the claimed “*communications network*.” Alternatively, the entire system depicted in Feroz’s Fig. 1 (described as a “network environment” and including networks 40, 50) renders obvious the claimed “*communications network*.” See Feroz at 5:39-40.

59. Feroz also explains that the application traffic management device 130 (including its packet processor 92, see Fig. 2) processes header fields, such as a layer 2 MAC address or layer 3 IP addresses:

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. In one embodiment, each traffic class has at least one attribute defining the criterion(ia) used for identifying a specific traffic class.

Feroz at 11:15-26.

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). In one embodiment, packet processor 92 **analyzes the source and destination IP addresses in the packet header** and scans host database 134 for matching entries.

Feroz at 18:3-9.

60. Feroz explains that the application traffic management device 130 includes at least two network interfaces, and potential more, that are used to receive and transmit packets. The interfaces may be, for example, Ethernet interfaces:

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.

Feroz at 6:26-45.

61. It would have been obvious to a POSITA from this description that the application traffic management device 130 would be forwarding packets (or, in terminology more often used with Ethernet, forwarding data frames). For example, Feroz shows in Fig. 1 that the application traffic management device 130 is on the data path between router 22 and network 40. Forwarding Ethernet frames (or packets) is a function that was commonly associated with Ethernet network switches. *See, e.g.,* US 2003/0174649, [0029] (“...a network switch, such as packet forwarding device...”). Especially for Feroz’s contemplated examples where the application traffic management device 130 would have three or more network interfaces, it would have been obvious to a POSITA that the application traffic management device 130 would have to decide how to forward each received packet, since there would be at least two potential destination interfaces. Based on Feroz’s overall description, it would have been obvious to a POSITA for the application traffic management device 130 to be a “*network switching element*.” This understanding is consistent with the description of “switches” in the ’884 patent:

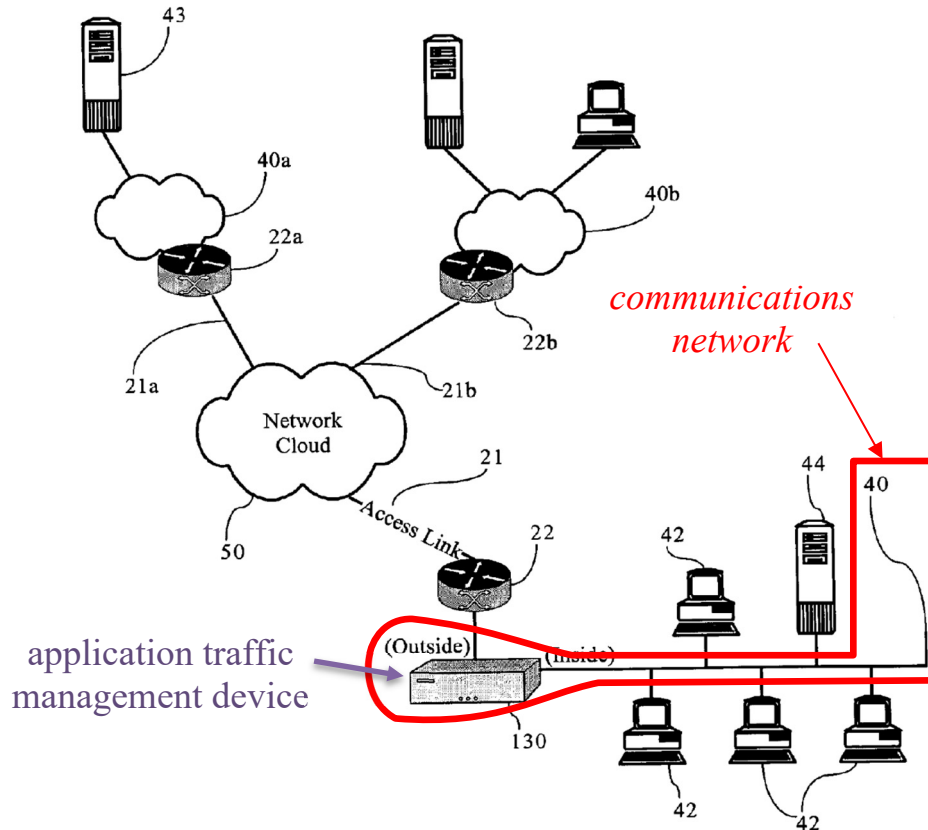
The switches may be configured to **process and route data (or data packets) at the data link layer (layer 2) of the OSI model, and/or process data (or data packets) at the network layer (layer 3)** (i.e., the switches may be considered “multilayer switches”). The switches may employ multiple connection interfaces in order to allow different user agents and/or user devices to connect to a

communications network, including Ethernet, Fibre Channel, ATM, ITU-T, G.hn, 802.11, and/or other like connection interfaces. According to various embodiments, where the **switches utilize L2/L3 connections to connect two or more devices at the datalink layer (L2) and/or the network layer (L3)** of the OSI model, the L2/L3 connections may operate as an enterprise private network, virtual private network, local area network (LAN), or other like computer network.

'884 Patent at 8:11-25.

62. Alternatively, a POSITA would have found it obvious to combine the application traffic management device 130 with router 22. Notably, the inventor Feroz described in another, similar patent application that “device 130 may be contained in router 22.” US7869366, 4:43-48. Integrating application traffic management device 130 and router 22 would directly connect the application traffic management device 130 with access link 21.

63. In summary, application traffic management device 130 at the edge of network 40 corresponds to the claimed “*network switching element in a communications network.*”



Feroz at Fig. 1 (annotated).

“...the network switching element including a target port...”

64. As I just discussed, the application traffic management device 130 includes two or more network interfaces, such as network interfaces 71 and 72 shown in Fig. 2:

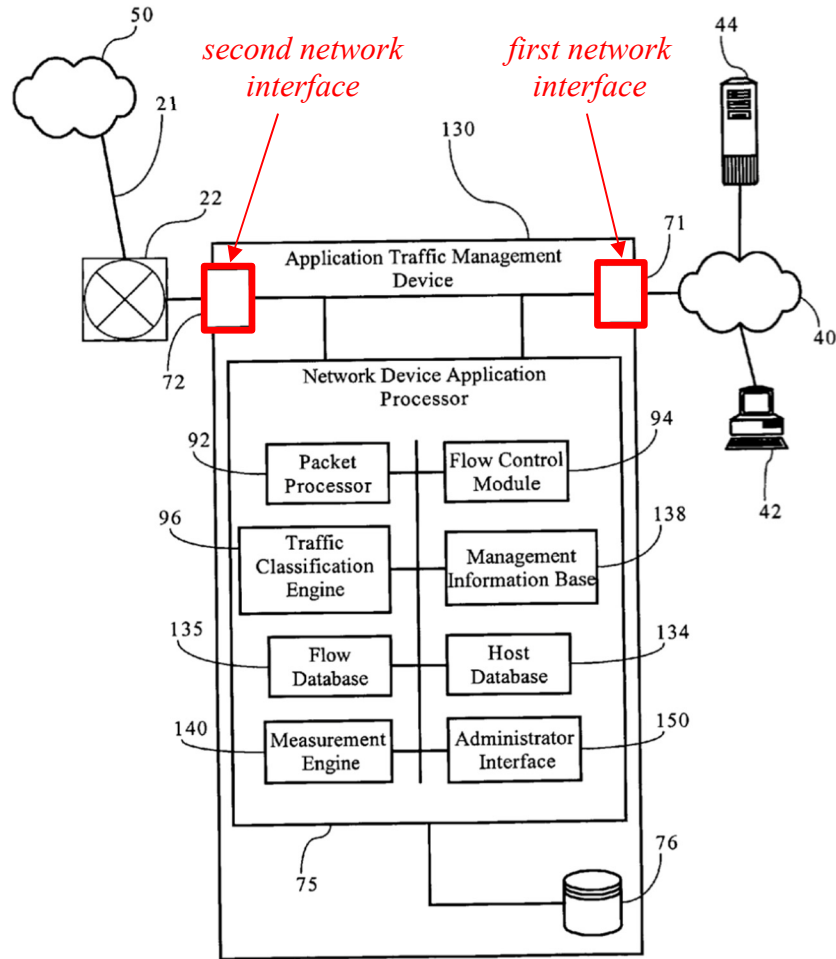


Fig._2

Feroz at Fig. 2 (annotated)

65. Either of the network interfaces 71, 72 is a “*target port*.” For network traffic flowing from network 50 to a device on network 40 (such as server 44 or client 42), packets would be received at network interface 72 and transmitted by network interface 71; thus, network interface 71 would be the “*target port*.” I have annotated this scenario on Fig. 2 below. For network traffic flowing in the opposite direction, network interface 72 would be the “*target port*.”

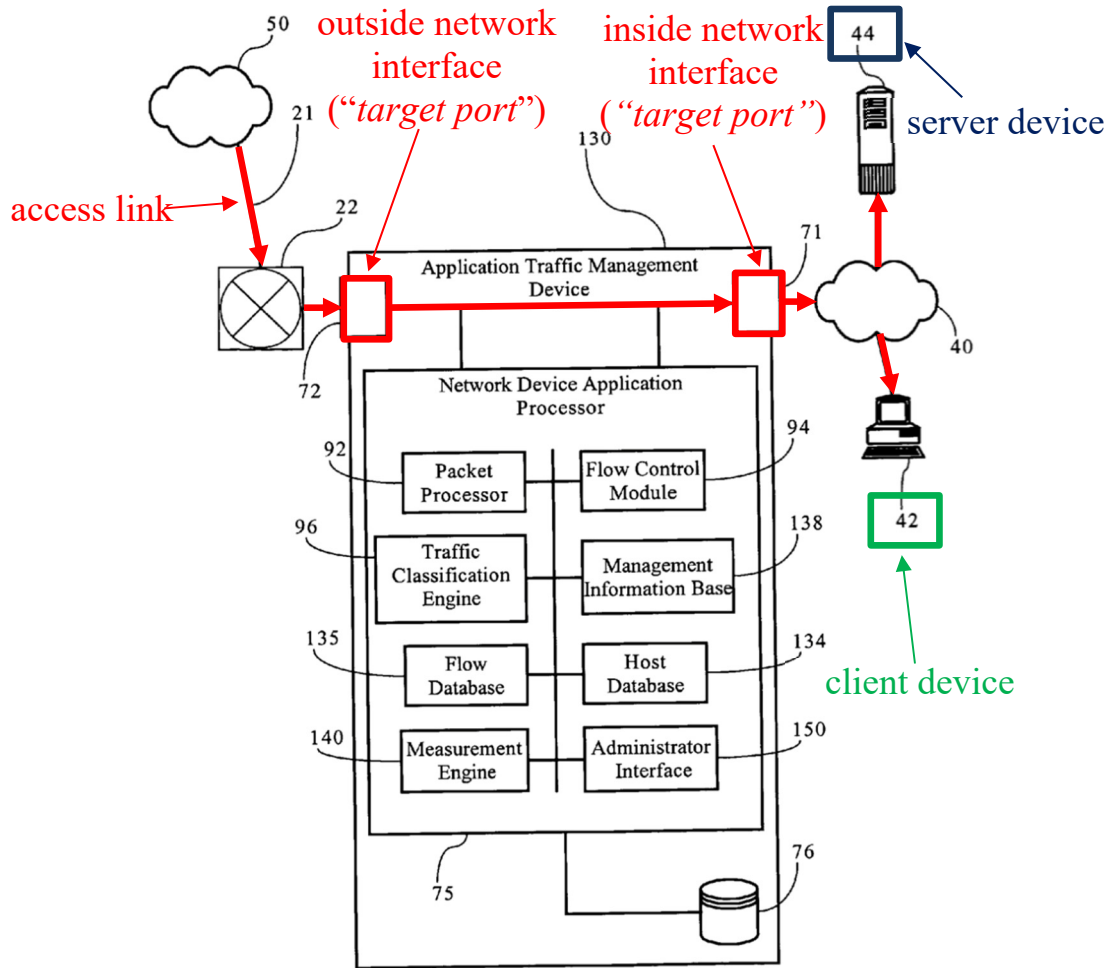


Fig. 2

Feroz at Fig. 2 (annotated).

66. In summary, Feroz’s disclosure of a network application traffic management device 130 (“a network switching element”) that dynamically adjusts bandwidth allocated to data flows traversing a network interface (“a target port”) renders obvious, “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.”

- b. [1.1] *monitoring, by the network switching element, a data flow traversing the target port of the network switching element;*

67. Feroz renders this limitation obvious.

68. Feroz describes that network application traffic management device 130 (“*network switching element*”) includes a network device application processor 75 that includes a flow control module 94, as depicted in Figure 2:

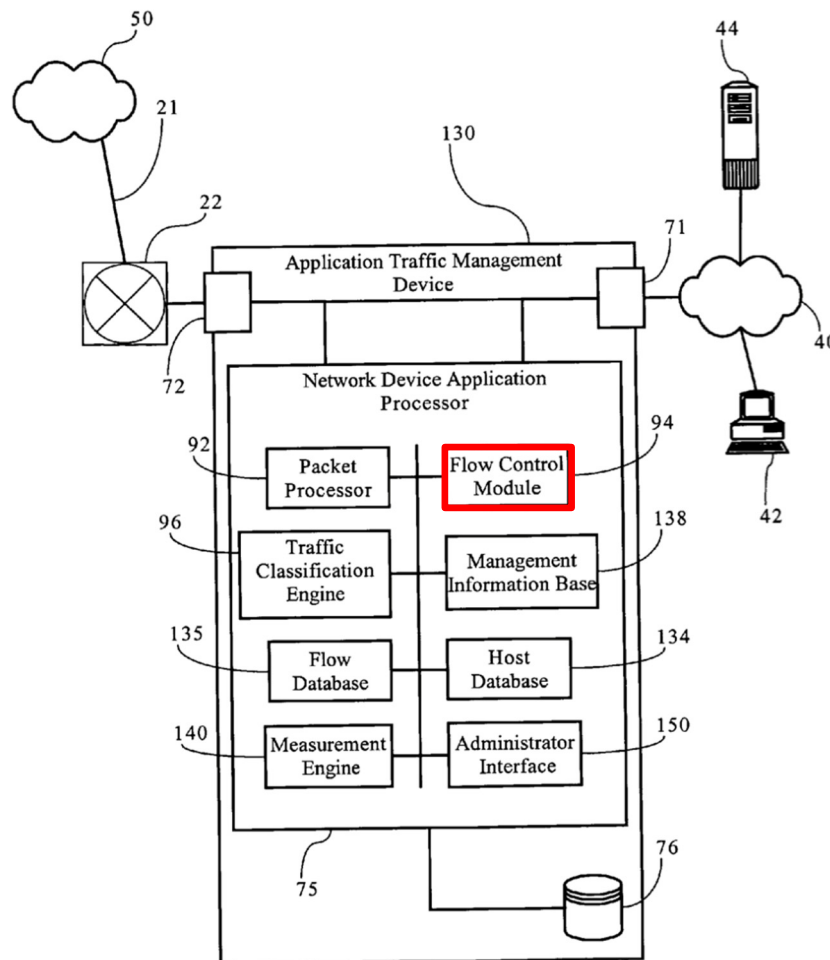


Fig. 2

Feroz at Fig. 2 (annotated).

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 as source and destination addresses, and the like) and storing packet attributes in a buffer structure, and maintaining one or more flow variables or statistics (such as packet count, current rate, etc.) in connection with the data flows and/or the source/destination hosts. The traffic classification engine 96, as discussed more fully below, is operative to classify data flows based on one or more attributes associated with the data flows. Traffic classification engine 96, in one implementation, stores traffic classes associated with data flows encountered during operation of application traffic management device 130, as well as manually created traffic classes configured 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.

Feroz at 6:51-7:9.

69. In one example, Feroz discloses that flow control module 94 “monitors”
bandwidth demand of each partition of access link 21:

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.

Feroz at 13:7-38.

70. In particular, Feroz describes that the demand for bandwidth at a given partition can be evaluated by “monitor[ing]” a “per-flow rate demand” for each data flow in the partition:

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.

Feroz at 13:7-38.

71. A POSITA would have found it obvious that Feroz's "packets in a flow" correspond to a "*data flow*" because packets contain data, as Feroz acknowledges elsewhere by describing "data packets" and detecting "new data flows." Feroz at 6:57-59 ("...packet processor 92 is operative to process data

packets, such as detecting new data flows...”) & 13:7-9 (“...data flows traversing access link 21...”).

72. Feroz further explains that the application traffic management device 130 maintains flow-related statistics, such as the flow rate and packet count:

As FIG. 3 illustrates, application traffic management device 130 may also perform other operations in response to arrival of the packet.... 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.

Feroz at 19:4-22.

73. More specifically, Feroz describes how the application traffic management device 130 includes a packet processor 92 that maintains “flow variables or statistics”:

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.... 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 as source and destination addresses, and the like) and storing packet attributes in a buffer structure, and **maintaining one or more flow variables or statistics**

(such as packet count, current rate, etc.) in connection with the data flows and/or the source/destination hosts.

Feroz at 6:51-65.

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).

Feroz at 18:28-31.

74. Feroz describes the overall packet processing actions of the packet processor 92 at 17:64-18:64. It would have been obvious to a POSITA for the packet processor 92 to be “*monitoring... a data flow*” since, as Feroz describes, the packet processor 92 must update the flow object as each new packet in a flow is received (e.g., by incrementing the packet count).

75. Thus, Feroz’s disclosure that application traffic management device monitors packets in a flow and maintains flow-based statistics renders obvious “*monitoring, by the network switching element, a data flow.*”

“...traversing the target port of the network switching element...”

76. Feroz similarly explains that “application traffic management device 130 is operative to classify and **manage data flows traversing** access link 21.” Feroz at 5:65-67. To do this, Feroz explains that “application traffic management device 130 processes **packets traversing it.**” Feroz at 17:41-42. As Feroz illustrates

in Fig. 2 (annotated below), application traffic management device 130 has two or more network interfaces; thus, for a packet to “traverse” application traffic management device 130, it would be obvious for the packets to be received at one network interface and then be transmitted by the other network interface. The packets in a data flow would “traverse” both the inbound and outbound network interfaces. As discussed at [1.0], either of Feroz’s network interfaces shown in Fig. 2 corresponds to the claimed “*target port*.” Thus, it would have been obvious to a POSITA from Feroz’s description that application traffic management device 130 performs its monitoring actions (discussed above) on one or more data flows that traverse a network interface (“*traversing the target port of the network switching element*.”

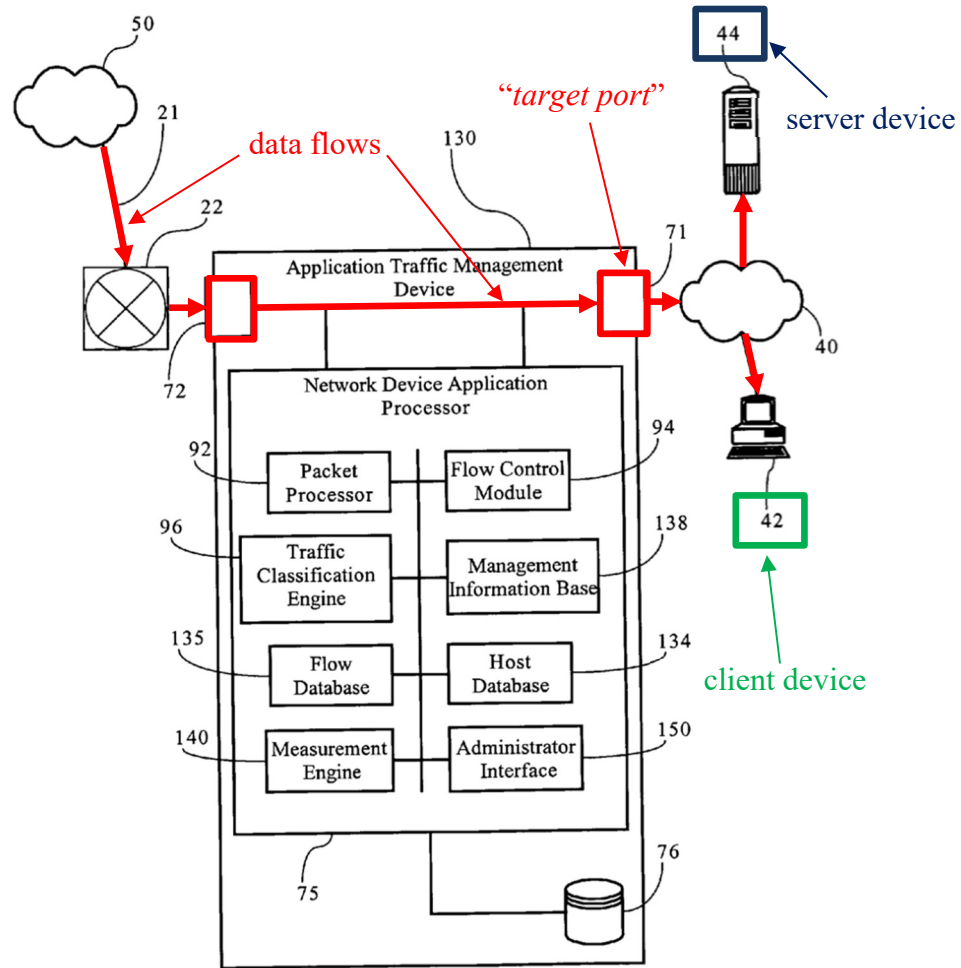


Fig._2

Feroz at Fig. 2 (annotated)

77. Feroz further confirms that the monitoring actions are performed as packets flow through the application traffic management device 130. For example, Feroz explains monitoring occurs as packets are “received” by application traffic management device 130:

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.

Feroz at 13:18-22.

78. Similarly, Feroz explains that monitoring occurs as packets are transmitted by the application traffic management device 130:

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.

Feroz at 19:18-22.

79. As I discussed at [1.0], the application traffic management device 130 transmits and receives packets via its network interfaces 71 and 72. Thus, it would have been obvious to a POSITA for the packet-monitoring operations described in Feroz to be applied to packets in a flow that traverse one of the network interfaces 71, 72 (“*target port*”).

80. A POSITA would have been motivated to monitor inbound data flows flowing from network interface 72 to network interface 71 towards a communications network 40 as described in Feroz to “control inbound traffic and reduce the inefficiencies associated with dropped packets.” Feroz at 3:18-22. To control the traffic, it would have been obvious to monitor the traffic. For example, monitoring inbound data flows allows traffic policies, such as a minimum

guaranteed bandwidth and/or maximum bandwidth, to be applied to a particular data flow involving a particular user or network application. Feroz at 3:39-43. In another example, monitoring inbound data flows could be used to protect or cap bandwidth for “all” traffic associated with an application, or controlling, prioritizing, protecting, and/or capping bandwidth for a particular user. Feroz at 3:43-47.

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 [*sic*] 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.

Feroz at 3:18-54.

81. In summary, Feroz's disclosure of application traffic management device 130 monitoring data packets in a flow across either network interface 72 or network interface 71 renders obvious "*monitoring, by the network switching element, a data flow traversing the target port of the network switching element.*"

- c. **[1.2] *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;***

82. Feroz renders this limitation obvious.

83. As discussed in [1.1], application traffic management device 130 (“*the network switching element*”) includes a flow control module 94 that monitors a “per-flow” rate demand for each data flow traversing network interfaces 71 and 72 (either of which is “*the target port*”).

84. The flow control module 94 also includes a “per-flow target rate” assignment mechanism to allocate bandwidth to each data flow. Feroz also refers to the per-flow target rate as a “per-flow bandwidth allocation”:

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.

Feroz at 4:46-61.

A.3.d. **Computing Per-Flow Bandwidth Allocation**

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.

Feroz at 20:5-9.

85. The per-flow target rate, also called a per-flow bandwidth allocation, corresponds to the claimed “*bandwidth that is currently allocated for the data flow*” because Feroz describes how the per-flow target rate is periodically recomputed, meaning that the computed value is always current:

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).

Feroz at 22:17-23.

86. Feroz's Figure 4B provides one example where flow control module 94 computes a "Flow Target Rate" (steps 334, 340, and 349):

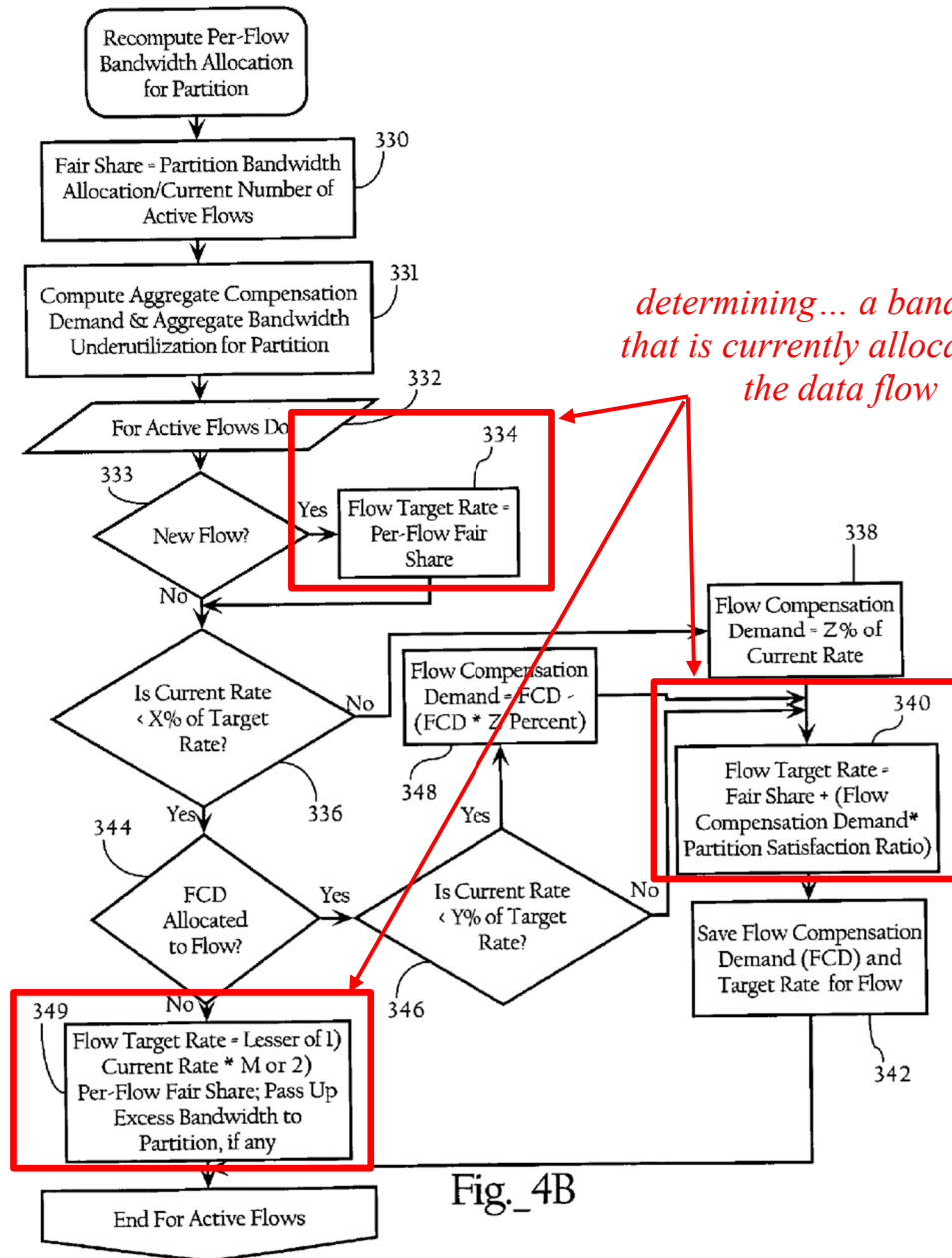


Fig. 4B

Feroz at Fig. 4B (annotated)

87. As shown in Figure 4B, a “Flow Target Rate” is computed for each active flow in a partition. Feroz explains that the Fig. 4B process is performed by the flow control module 94, which is part of the application traffic management device 130 (*“the network switching element”*). Feroz at 20:22-26 (*“...as FIG. 4B illustrates, flow control module 94 computes...”*), Fig. 1. Thus, Feroz renders obvious the step of *“determining, by the network switching element, ... a bandwidth that is currently allocated for the data flow.”*

88. As I mentioned previously, it would have been obvious to a POSITA that the computed flow target rate is a “bandwidth that is **currently** allocated” because Feroz describes recomputing the flow target rate periodically. Feroz also describes how the “per-flow target rate is enforced,” which may be by using “TCP Rate Control technologies.” Feroz, 19:61-20:4. It would have been obvious to a POSITA for the bandwidth allocation being enforced by Feroz’s application traffic management device 130 to be a “*current*[.]” bandwidth allocation. Feroz emphasizes, for example, how frequently make dynamic adjustments to per-flow bandwidth allocations can “improve... network application performance.” Feroz at 7:24-28. Making dynamic adjustments (and thus enforcing “*currently allocated*” per-flow target rates) allows the application traffic management device 130 to be “*respons*[ive] to network traffic demands.” Feroz at 14:56-58. These disclosures further confirm that a POSITA would have understood that Feroz discouraged using

stale or outdated bandwidth allocations. The per-flow target rates that Feroz's application traffic management device 130 would have obviously been those "*current allocated*," as claimed.

"...a bandwidth allocation for the target port..."

89. Feroz describes applying its per-flow rate control processes "at any point prior to egress from application traffic management device 130," including for example by modifying packets being transmitted:

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 TCPACK 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.

Feroz at 19:59-20:4.

90. Packets would be "transmitted" by the network interfaces of the application traffic management device 130. The transmission of a packet by a

network interface also corresponds to the point at which the packet would “egress” from application traffic management device 130. Thus, it would have been obvious to a POSITA for Feroz’s per-flow rate controls to be enforced at either or both of the network interfaces 71 and 72 (“*the target port*”), and therefore for the enforced per-flow target rate to be a “*bandwidth allocation for the target port.*”

91. Feroz’s discussion of partition-based flow control techniques reinforces the obviousness of enforcing a “*bandwidth allocation for the target port.*” Feroz explains that a partition may correspond to either of the network interfaces:

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).

Feroz at 17:9-17.

92. As discussed further below regarding limitations [1.3]-[1.5], Feroz’s per-flow target rates are part of Feroz’s overall partition-based bandwidth allocation scheme, and therefore it would have been obvious for the per-flow target rate to

similarly “correspond to... a given network interface.” Feroz at 17:14-15. Thus, it would have been obvious to a POSITA for Feroz’s per-flow target rate to be a bandwidth allocation for a data flow at “*the target port*,” as claimed.

d. [1.3] *determining, by the network switching element, a fair-share bandwidth allocation for the target port,*

93. Feroz renders this limitation obvious.

94. First, as discussed in the previous limitations, Feroz describes an application traffic management device 130 that enforces “bandwidth utilization controls to data flows traversing access link 21.” Feroz at 13:7-17. Feroz illustrates in Fig. 1 that the application traffic management device 130 is not itself directly coupled to access link 21 (due to intervening router 22). But since router 22 only has two connections (to access link 21 and to application traffic management device 130), it would have been obvious to a POSITA that all or substantially all the data flows traversing access link 21 would also traverse at least outside network interface 72 of application traffic management device 130. This is because the figure does not show any other devices where packet traffic across access link 21 would originate or terminate. Notably, Feroz does not describe router 22 as itself doing anything other than being part of the communication path between network 40 and network 50 (that is, providing connectivity between application traffic management device 130 and access link 21):

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.

Feroz at 5:49-53; *see also* Feroz at 6:4-9.

95. Thus, it would have been obvious to a POSITA for there to be either no data flows originating or terminating at router 22, or for the bandwidth used by such data flows to be negligible. As a result, all (or substantially all) of the data flows traversing access link 21 would also traverse network interfaces 71 and 72, thereby allowing application traffic management device 130 to apply bandwidth utilization controls to data flows traversing access link 21 (as Feroz contemplates) by controlling those data flows at network interface 71 or 72.

96. Second, Feroz describes employing a “partition-based” bandwidth allocation scheme that monitors bandwidth demand and compares it to a “fair share allocation”:

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 band width exceeds its weighted fair share allocation.

Feroz at 13:7-14.

97. This requires that the flow control module 94 compute “weighted, fair share **bandwidth allocations...** for the partitions”:

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.

Feroz at 14:46-49.

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. Flow control module 94 then determines whether to push the flow queue onto the high priority (60b) or low priority (60a) partition queue corresponding to the identified partition.

Feroz at 19:28-50.

As FIG. 4B illustrates, if the current rate of the flow is greater than or equal to X percent of its target rate (336), flow control module 94 computes a flow compensation demand based on Z percent of the current rate (338). **Flow control module 94 assigns a flow target rate equal to the per-flow fair share bandwidth allocation**, plus the compensation demand for the flow multiplied by the partition satisfaction ratio (340). In one implementation, the partition satisfaction ratio is the aggregate bandwidth utilization divided by the aggregate compensation demand. Flow control module 94 then saves the flow compensation demand and the target rate for the flow in the corresponding flow object (342).

Feroz at 21:16-27.

98. Feroz's partition-based fair-share bandwidth allocations correspond to the claimed "*fair-share bandwidth allocation for the target port*" in multiple different ways. I will discuss and explain them individually.

"for the target port": Bandwidth control for all traffic flowing through the "target port" in one direction

99. As one example of a "partition-based" bandwidth allocation scheme, Feroz explains that a partition may correspond to all traffic passing in one direction through one network interface, for example, all traffic at network interface 71 that is

destined for network interface 72—in other words, all traffic inbound to network interface 71:

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).

Feroz at 17:12-17.

In one implementation, where only leaf partition nodes are scheduled for egress from application traffic management device 130, each leaf traffic class must be associated with a leaf partition node in the partition configuration hierarchy. 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.

Feroz at 14:9-18.

100. Riddle provides further description of the use of partitions. As I noted previously, Riddle is incorporated by reference into Feroz. Riddle provides as a partitioning example the use of a separate partitions for traffic flowing in each

direction. Thus, Riddle contemplates having one partition for all inbound traffic and another partition for all outbound traffic:

In one embodiment, at the highest level, **a partition exists for all available outbound bandwidth**, while **another partition exists for all available inbound bandwidth** across the particular access link. These partitions are then sub-dividable to form a hierarchical tree.

Riddle at 6:58-61.

101. It would have been obvious to a POSITA to monitor and control bandwidth utilization as contemplated by Feroz and Riddle, that is, by treating inbound and outbound data flows separately by having just one inbound bandwidth partition and one outbound bandwidth partition for each network interface. A POSITA would have found such an arrangement to be entirely sensible because most network communication technologies supported full-duplex communication, which allows data to be transmitted and received simultaneously. With full-duplex communication, there is no competition between transmission and reception activities, and hence, no need to allocate any shared bandwidth resources between them. Since Feroz describes the network interfaces 71 and 72 as being, for example, Ethernet network interfaces, and since most Ethernet network interfaces at the time supported full-duplex communication, it would have been appropriate for the

application traffic management device 130 to have separate partitions for inbound and outbound traffic.

102. Where there is only a single outbound partition for a network interface, it would have been obvious to a POSITA for Feroz's process to compute a fair share bandwidth allocation for that single outbound partition. Likewise, for a single inbound partition for a network interface, it would have been obvious to a POSITA for Feroz's process to compute a fair share bandwidth allocation for that single inbound partition. Thus, the combination of Feroz and Riddle renders obvious monitoring and controlling bandwidth usage for all inbound data flows (or, separately, all outbound data flows) traversing a network interface of the application traffic management device 130.

Bandwidth control for the “*target port*” as a whole

103. If this limitation is argued or interpreted as requiring a fair share bandwidth allocation for the “*target port*” as a whole (encompassing both the transmission and reception of data), then Feroz and Riddle still render the limitation obvious. Feroz contemplates having a single partition that includes both inbound and outbound data flows:

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.**

Feroz at 14:14-20.

104. As I discussed above, most network interface communication technologies supported full-duplex communication, for which aggregating inbound and outbound data flows into a single partition would not make sense. However, there also existed some half-duplex network interface technologies. The original version of Ethernet, for example, was a half-duplex communication technology. Indeed, an early specification document for Ethernet expressly states that full duplex operation is not a goal for the technology. *See* DIX Ethernet (Ex.1014) at 6. Even into the 1990s, new Ethernet variants were being introduced that supported only half-duplex operation. *See, e.g.,* Breyer (Ex.1010) at 68-69 & 72 (discussing 100BASE-T4, where “full-duplex operation is not possible”). While Feroz explains that the network interfaces on the application traffic management device 130 may use Ethernet, Feroz does not identify any specific implementation or variant of Ethernet. Thus, it would have been obvious to a POSITA for the network interfaces to use any known Ethernet variant, including those that supported only half-duplex operation. In that scenario, it would have been obvious to use a single partition for bandwidth allocation because the available bandwidth must be shared between

transmission and reception. In other words, any time spent transmitting data cannot be used for receiving data, and vice versa. Thus, the network interface's available bandwidth is shared between inbound and outbound data flows.

105. In summary, Feroz and Riddle render obvious allocating a fair share bandwidth to a partition, where the partition corresponds to the entire available bandwidth for a network interface ("*target port*"). These ideas render obvious "*determining, by the network switching element, a fair-share bandwidth allocation for the target port*" as claimed.

Bandwidth control for a group of data flows
flowing through the "*target port*"

106. I have been informed that the '884 Patent Owner may assert that this claim limitation pertains to determining a fair share bandwidth allocation for a group of data flows that traverse the "*target port*." The combination of Feroz and Riddle render such an interpretation obvious.

107. Feroz explains that a partition may correspond to an aggregation of data flows, for example, multiple data flows that share characteristics such as a traffic class, application, or user:

A partition operates to manage bandwidth for aggregate data flows associated with a traffic class.

Feroz at 13:54-55.

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).

Feroz at 3:37-47.

108. Where a partition corresponds to data flows that have been aggregated according to a traffic class, on an application basis, or on a user basis, Feroz's partition-based bandwidth allocation processes would determine a fair share bandwidth allocation for the group of related data flows traversing a network interface. Under the claim interpretation that the '884 Patent Owner may advance, this disclosure renders the claim limitation obvious.

Bandwidth control for an individual data flow
flowing through the "target port"

109. I have been informed that the '884 Patent Owner may assert that this claim limitation pertains to determining a fair share bandwidth allocation for an

individual data flow that traverses the “*target port*.” The combination of Feroz and Riddle render such an interpretation obvious.

110. Feroz describes how fair share allocations may be provided down to the individual flow level:

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.

Feroz at 14:46-52.

In addition, as discussed more fully below, **each data flow corresponding to a given partition is allocated** a portion of that partition’s bandwidth allocation **based on a fair share algorithm** discussed below.

Feroz at 16:44-47.

111. Since Feroz describes guaranteeing a “fair share of the bandwidth” to an individual data flow, it would have been obvious to a POSITA for Feroz’s application traffic monitoring device 130 to compute that “fair share” bandwidth value. Feroz describes in conjunction with Fig. 4B, for example, a method of “computing a per-flow fair share allocation.” Feroz at 20:7-8. That per-flow fair

share allocation is used, for example, in the process discussed above for limitation [1.2] of calculating a current “target rate” for a data flow.

112. Thus, under a single-flow claim interpretation that the ’884 Patent Owner may advance, Feroz and Riddle still render the claim limitation obvious.

Conclusion for [1.3]

113. As discussed above, Feroz and Riddle render obvious “*determining... a fair share bandwidth allocation for the target port*” in multiple ways. Regardless of whether this limitation is interpreted as requiring a “*fair share bandwidth allocation for the target port*” at the level of (1) the port overall (including bidirectional data flows), (2) data flows through the port in one direction, (3) a group of related data flows through the port, or (4) an individual data flow through the port, the limitation would have been obvious.

- e. **[1.4] *the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and***

114. Feroz renders this limitation obvious.

115. Feroz describes allocating a “fair share” of bandwidth based on user-assigned weights for each partition:

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.

Feroz at 14:21-31.

116. Feroz provides a more detailed explanation of this weight-based allocation of the fair share bandwidth. It is, effectively, a weighted pro rata allocation of the bandwidth to each partition:

In one implementation, starting from the root node, the weights of the direct child nodes (here, A & B) are compared to allocate available bandwidth of the root partition. The band width allocation that each node receives, in one implementation, is based on the following equations:

Aggregate Weight= Σ Weights from Child Nodes

$$\text{Allocation Per Partition } (c) = \frac{\text{Parent Allocation} * \text{Configured Weight of Partition } (c)}{\text{Aggregate Weight}}$$

Feroz at 14:67-15:14.

117. Feroz's equation above includes the term $\frac{\text{Configured Weight of Partition (c)}}{\text{Aggregate Weight}}$ which represents a "*proportion[]*" used to allocate to a specific child partition (c) a portion of the bandwidth available in a parent partition.

118. Feroz contemplates there being one or more "root" partitions that have no parent. The allocation to child partitions begins "starting from the root node." Feroz at 15:1. The "Root partition node 82 represents the total capacity of access link 21." Feroz at 16:57-58. The total capacity of access link 21 represents a "total bandwidth" of the application traffic management device 130 at outside network interface 72, since all data flows that traverse access link 21 would also traverse outside network interface 72.

119. In short, Feroz describes making a pro rata allocation of a fair share bandwidth allocation to a partition, which renders obvious "*the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element.*" I provide below further comments and details regarding how this limitation is rendered obvious under any of the potential mappings of "*target port*" discussed above at [1.3].

"proportional": Bandwidth control for the "target port" as a whole

120. As I discussed for limitation [1.3], Feroz describes a single root node partition that applies to "both inbound and outbound data flows" and "represents the

total capacity of access link 21.” Feroz at 14:19-20, 16:57-58. Since there is just one partition (and no other partitions to share bandwidth with), it would have been obvious to a POSITA for the single root node partition to be allocated 100% of the fair share bandwidth (“*a proportional allocation*”). Since the bandwidth of access link 21 is managed by the application traffic management device 130, it would have been obvious to a POSITA that the “total capacity of access link 21” represents a “*a total bandwidth of the network switching element*.” Thus, Feroz renders obvious this limitation.

121. Additionally, as I discussed above for limitation [1.3], it would have been obvious for the application traffic management device 130 to have a network interface that supports only half-duplex operation. In that situation, it would have been obvious to a POSITA that the total communication bandwidth would be split between transmission capacity and reception capacity. To facilitate this, Feroz describes “separate partition configurations... for data flows traveling in the inbound or outbound directions.” Feroz at 14:14-20. Feroz explains that a user would provide weights to control the splitting of the root partition’s fair share bandwidth between these child partitions. Feroz at 14:21-31. For example, a user might choose to divide the fair share bandwidth equally between transmission and reception by providing their corresponding partitions with equal weights. Or the user might choose to give a greater portion of the fair share bandwidth to either transmission or to reception.

Regardless of the specific weights assigned by the user, Feroz's technique of dividing fair share bandwidth from a parent partition to its child partitions would provide "*a proportional allocation*" as claimed.

**"proportional": Bandwidth control for all traffic
flowing through the "target port" in one direction**

122. As I discussed above at limitation [1.3], Feroz and Riddle describe having two separate, root-level partitions for inbound and outbound communications. *See* Feroz at 14:16-18; Riddle at 6:58-61. This arrangement would be appropriate for a full-duplex network interface where data transmission and data reception occur independently of one another (including simultaneously).

123. Riddle explains that two such partitions exist at the highest level:

In one embodiment, at the highest level, a partition exists for all available outbound bandwidth, while another partition exists for all available inbound bandwidth across the particular access link. These partitions are then subdividable to form a hierarchical tree.

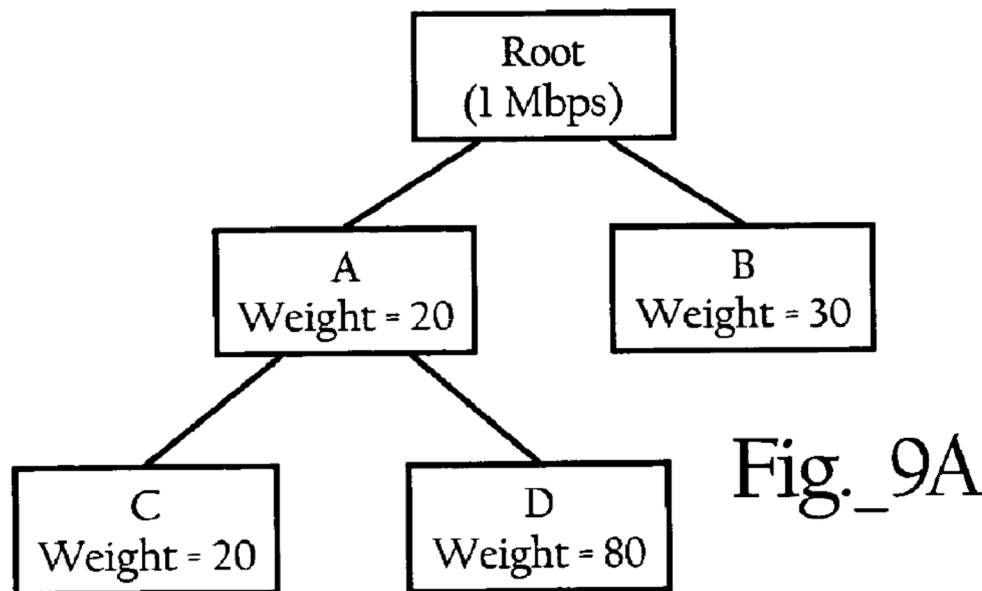
Riddle at 6:58-61.

124. Since each of these "highest level" partitions would not have to share bandwidth with each other, it would have been obvious to a POSITA for them to be allocated 100% of the fair share bandwidth in their respective communication directions. That is, it would have been obvious for the inbound partition to be allocated 100% of the fair share bandwidth for data reception, and for the outbound

partition to be allocated 100% of the fair share bandwidth for data transmission. These allocations are “*proportional*” since each partition would be entitled to the entire available fair share allocation in their respective communication direction.

“proportional”: Bandwidth control for a group of data flows flowing through the “target port”

125. As I discussed at [1.3], Feroz and Riddle describe having separate partitions for different types of traffic. For example, traffic flows can be aggregated together and treated as a group when they relate to a traffic class, to a particular application, or to a particular user. A group of traffic flows are assigned to a single partition. Then, user-assigned weights are used to allocate fair share bandwidth among the partitions, with the partitions arranged in a hierarchy. Feroz illustrates this concept in Fig. 9A:



Feroz at Fig. 9A.

126. Fig. 9A shows a root partition node that has a bandwidth of 1 Mbps (megabit per second). Feroz's fair share bandwidth allocation is then used to compute the fair share bandwidth allocated on a proportional basis to partition A:

$$\begin{aligned} & \text{Root bandwidth} * \frac{\text{Weight of Partition A}}{\text{Weight of Partition A} + \text{Weight of Partition B}} \\ &= 1\text{Mbps} * \frac{20}{20 + 30} \\ &= 0.4 \text{ Mbps} = 400 \text{ Kbps} \end{aligned}$$

127. As the formula shows, the fair share bandwidth allocated to partition A is in proportion to the weight assigned to partition A. Similar calculations are made for the other partitions in Fig. 9A. Feroz provides the following table showing the resulting proportional allocations:

TABLE 1

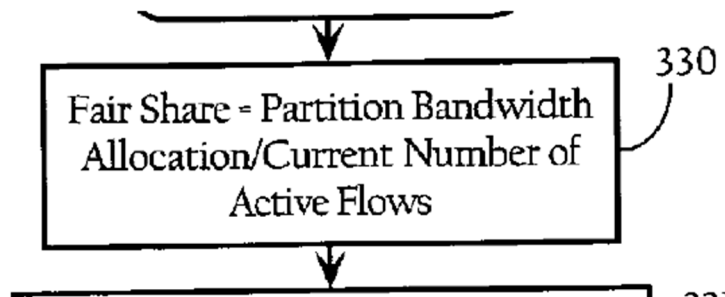
Partition	Weighted Fair share
A	400K
C	80K
D	320K
B	600K

Feroz at 15:15-26.

128. By using user-assigned weights to divide the fair share bandwidth of a parent partition among its child partitions, Feroz renders obvious a “*proportional allocation*” as claimed.

**“proportional”: Bandwidth control for an
individual data flow**

129. As I discussed for limitation [1.3], Feroz and Riddle describe determining a fair share bandwidth allocation for an individual data flow. Feroz's Fig. 4B process shows that this occurs at step 330:



Feroz at Fig. 4B (partial).

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).

Feroz at 20:9-11.

130. Feroz's per-flow fair share bandwidth allocation is "*proportional*" to the number of active flows within a given partition. The per-flow fair share bandwidth allocation is also based on the "current aggregate bandwidth allocated to the partition," which represents a "*total bandwidth*" associated with the application traffic management device 130. In summary, Feroz's per-flow fair share bandwidth allocation is a "*proportional allocation*" as claimed.

- f. **[1.5] *adjusting, by the network switching element, the bandwidth allocation for the target port based on the fair-share bandwidth allocation.***

131. Feroz renders this limitation obvious.

132. As discussed at [1.2], Feroz describes a “per-flow target rate” (also called a “per-flow bandwidth allocation”) that corresponds to the claimed “*the bandwidth allocation for the target port.*” Feroz describes “adjusting” the target rate for each flow:

This **per-flow target rate may be limited or adjusted** by
a per-flow rate policy configured by a user.

Feroz at 4:27-29.

133. Feroz also adjusting per-flow target rates based on observed bandwidth demands. The flow control module 94 (within the application traffic management device 130) determines how much, if any, of a given partition’s fair share allocation is going unused. Since the unused portion is part of the partition’s fair share bandwidth allocation, it would have been obvious that the unused portion is “*based on the fair-share bandwidth allocation*” for the partition. The unused portion is then reallocated to another partition as an excess bandwidth allocation:

FIGS. 8A and 8B together illustrate a method, according to one implementation of the invention, directed to computing aggregate bandwidth allocations across partitions in response to observed compensation demand and bandwidth underutilization. In one implementation,

flow control module 94 traverses the hierarchical partition configuration (402), starting at the leaf nodes, to **compute the un-utilized fair share** (UFS) and compensation demand (CD) for each partition node (402,404, 406 and 408) that is sent up to the parent partition node.

Feroz at 22:24-33.

Flow control module 94, in one implementation, then computes the excess bandwidth available for distribution. Flow control module 94, starting at the root level in the partition configuration hierarchy (420), computes the excess band width allocation available for distribution to child partitions for all non-leaf partitions in that current level (422). As FIG. 8B illustrates, if the aggregate compensation demand (CD) is less than the aggregate unutilized fair share (UFS) (424), flow control module 94 **sets the excess bandwidth allocation (BW) for the partition** node to the aggregate compensation demand (426) and sets the partition satisfaction ratio to one, since the excess bandwidth allocation (BW) is equal to the compensation demand (CD) for the partition.

Feroz at 22:53-65.

134. Feroz illustrates this bandwidth reallocation process in connection with Figs. 8A and 8B and Fig. 4B. At a high level, Figs. 8A and 8B describe determining an amount of unused bandwidth for a partition that is available to be reallocated elsewhere. Fig. 4B then describes allocating the excess bandwidth from the Fig. 8B

process to individual data flows in another partition. Specifically, step 330 uses the “current aggregate bandwidth allocated to the partition.” Feroz at 20:9-11. It would have been obvious that this “aggregate bandwidth” includes any excess bandwidth reallocated from another partition. Feroz confirms this understanding by referencing how “excess bandwidth” may be reallocated and incorporated into an “aggregate bandwidth”:

However, since the maximum bandwidth setting for Partition A is less than the weighted fair share allocation, **the excess bandwidth (100 Kbps) is redistributed among the sibling partitions, resulting in an aggregate bandwidth allocation** of: 1) Partition A-100 Kbps; 2) Partition B 350 Kbps; and Partition C-550Kbps.

Feroz at 16:22-27.

135. As discussed for limitation [1.2], Feroz’s step 330 calculates a per-flow target rate, which corresponds to the claimed “*bandwidth allocation for the target port.*” As I discussed previously, the Fig. 4B process (including step 330) is performed periodically. Thus, for the second iteration through Fig. 4B (and every iteration afterwards), the re-computation of the per-flow target rate corresponds to “*adjusting*” the per-flow target rate. Thus, Feroz renders obvious “*adjusting... the bandwidth allocation*” as claimed.

136. The bandwidth that Feroz describes reallocating is a portion of a partition's fair share bandwidth. Thus, the reallocation is "*based on the fair-share bandwidth allocation*" as claimed.

137. As I discussed at [1.0], Feroz's application traffic management device 130 corresponds to the claimed "*network switching element*." The bandwidth reallocation processes of Figs. 4B, 8A, and 8B are performed by flow control module 94, which is part of the application traffic management device 130:

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.**

Feroz at 19:8-12.

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.

Feroz at 20:46-49.

FIGS. 8A and 8B together illustrate a method, according to one implementation of the invention, directed to computing aggregate bandwidth allocations across partitions in response to observed compensation demand and bandwidth underutilization. In one implementation,

flow control module 94 traverses the hierarchical partition configuration (402), starting at the leaf nodes, to compute the un-utilized fair share (UFS_P) and compensation demand (CD_P) for each partition node (402, 404, 406 and 408) that is sent up to the parent partition node.

Feroz at 22:24-33.

As **FIG.8B** further illustrates, **flow control module 94** then computes the partition satisfaction ratio for the partition node (430).

Feroz at 23:8-10.

138. Thus, Feroz renders obvious that “*adjusting*” is performed by “*the network switching element.*”

2. Claim 17

139. Claim 17 is substantially the same as claim 1, except that claim 17 is drafted as a device claim (directed to an “*edge switch*”) instead of a method. Because claim 17 recites essentially the same subject matter as claim 1, claim 17 would have been obvious to a POSITA for the same reasons I provided above for claim 1. Even so, I will discuss below the specific, minor wording differences between the two claims.

- a. **[17.0] *An edge switch for adjusting bandwidth allocation in a communications network, the edge***

***switch including a target port, the edge switch
configured to:***

140. Feroz renders this limitation obvious.

141. As discussed in [1.0], Feroz’s application traffic management device 130 renders obvious a “*network switching element*.”

142. Feroz describes how the application traffic management device 130 “is deployed **at the edge of network 40**.” Feroz at 5:62-64. It therefore would have been obvious to a POSITA for the application traffic management device 130 to be an “*edge switch*.”

143. Every step of claim 1 is recited as being performed by the *network switching element*.” My analysis of claim 1 showed that it would have been obvious for Feroz’s application traffic management device 130 to perform each of those steps. For the same reasons, it would have been obvious for application traffic management device 130 to be “*configured to*” perform those steps as they are recited in claim 17. For example, Feroz expressly states that the application traffic management device 130 is configurable, and it would have been obvious for the application traffic management device 130 to be “*configured*” through the use of software modules.

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.

Feroz at 6:9-20.

Application traffic management device 130 can be configured to include matching rules that define a plurality of network applications commonly found in enterprise networks. Such as database applications, Citrix® flows, ERP applications, and the like.

Feroz at 11:45-49.

144. The rest of limitation [17.0] is substantially identical to language in limitation [1.0] and would have been obvious to a POSITA for the reasons I discussed previously for that limitation.

b. [17.1] *monitor a data flow traversing the target port;*

145. See analysis at [1.1].

a. [17.2] *determine 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;

146. See analysis at [1.2].

- a. [17.3] *determine a fair-share bandwidth allocation for the target port,*

147. See analysis at [1.3].

- a. [17.4] *the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and*

148. See analysis at [1.4]. I note that there is no earlier introduction of a “network switching element” in claim 17. For the purposes of my analysis, I assume that the “network switching element” recited here is supposed to refer to the “edge switch” introduced in the claim 17 preamble (limitation [1.0]). In my analysis of both [1.0] and [17.0], I mapped both these components to Feroz’s application traffic management device 130.

- b. [17.5] *adjust the bandwidth allocation for the target port based on the fair-share bandwidth allocation.*

149. See analysis at [1.5].

IX. PETERSEN, ALONE OR IN COMBINATION WITH PALACHARLA, RENDERS OBVIOUS CLAIMS 1 AND 17

A. Summary of Petersen

150. Petersen discloses a network switching system and a method for managing traffic loads from multiple virtual output queues (VOQs) to prevent congestion by adjusting transmission rates based on the capacity of shared resources.

Petersen at Abstract, [0006]. The network switching system includes a plurality of ingress and egress devices, such as ingress and egress ports, each having a plurality of VOQs for storing traffic to be forwarded to a destination, and one or more intermediary devices, such as switch fabric devices, coupled between the ingress and egress. Petersen at [0023], [0034]-[0036]. The ingress ports exchange queue state messages with each other to indicate the state of their VOQs, such as their level of fullness. Petersen at [0106]-[0107].

151. Based on the exchanged messages, Petersen's ingress ports determine the total traffic load on shared resource within the network switching system, such as a fabric switch or crossbar switch. Petersen at [0092], [0033]. The ingress ports also determine the capacity of the shared resource, which may be, for example, the bit rate or traffic capacity of the resource. Petersen at [0091].

152. The ingress ports then adjust the transmission rate from one or more of their active VOQs over the shared resource such that the total traffic load from the active VOQs does not exceed the capacity of the shared resource. Petersen at [0092]. The adjustment of the transmission rate may be based on a fair or weighted allocation of the capacity among the active VOQs, depending on the traffic priority or class of the VOQs or the traffic. Petersen at [0093], [0097]. The adjustment of the transmission rate may prevent or reduce congestion in the network switching system. Petersen at [0102]-[0103].

B. Summary of Paracharla

153. Paracharla describes a method of flexibly managing virtual queues of a switching system. Paracharla at Abstract, [0006]-[0007]. The method provides non-blocking support for data transmission between input and output ports of a switch, while reducing the memory requirements and costs of implementing a large number of virtual queues. Paracharla at [0006]-[0007]. Virtual queues are dynamically allocated from a central pool based on need. Paracharla at [0006]-[0007]. Thus, the Paracharla provides techniques for “queuing packets in per-flow queues.” Paracharla at [0004].

C. Reasons to Combine Petersen and Paracharla

154. Combining Petersen and Palacharla would have been obvious for multiple reasons.

155. Petersen describes using virtual output queues (VOQs) to organize and manage the transmission of packets across a shared resource, such as a switch fabric or crossbar switch. These queues are used, for example, to temporarily enqueue packets at an ingress port before the packets are transferred across the shared resource to an egress port. Petersen suggests that “there may be one VOQ in an ingress device per destination,” such as one queue per “egress port.” Petersen at [0036]. But this is just one approach, and Petersen is open to other arrangements. *See, e.g.*, Petersen at [0105] (describing having a “traffic priority (or traffic class)

for each queue”), [0033] (advising that Petersen’s expressly disclosed embodiments are “merely some example system configurations”), [0055] (“other configurations may be used”).

156. Paracharla describes another configuration of virtual output queues that was known in the art, namely, the use of “per-flow queues.” Paracharla at [0004]. This configuration was used, Paracharla explains, by “many switches.” Paracharla at [0004]. Since the use of per-flow queueing of packets was commonly used, it would have been obvious to a POSITA to combine Petersen and Paracharla’s teachings such that Petersen’s network switching system would implement per-flow queues as taught by Paracharla.

157. Another reason to combine Petersen and Paracharla would have been their common focus on preventing “head of line blocking.” Petersen at [0036]; Paracharla at [0003] (“a non-blocking port configuration may be beneficial”). Paracharla explains that the traditional approach to “non-blocking flows quickly becomes expensive,” since there would be potentially thousands of distinct queues. Paracharla at [0005]. But Paracharla describes memory-efficient techniques for creating non-blocking queues. Paracharla at [0036] (describing “efficient memory use” for queueing “across all output ports”). Thus, a POSITA would have been motivated to combine the teachings of Petersen and Paracharla to obtain the known benefits of non-blocking queueing and the benefits of reduced memory consumption

from Paracharla's "dynamically configured" virtual output queues. Paracharla at [0006]; *see also* [0003] ("a non-blocking port configuration may be beneficial"), [0053] ("virtual output queues may be updated dynamically").

158. Relatedly, I note that Petersen provides only a high-level description of its virtual output queues. Petersen at [0036]-[0037]. Paracharla provides a more detailed description of how to implement virtual output queues, for example, in the description of Figures 3 and 4 that explain how to structure and organize virtual output queues within a memory space using pointer fields, indexing, and linked lists. Paracharla at Figs. 3-4, [0026]-[0052]. Thus, a POSITA implementing the virtual output queues described in Petersen would have been motivated to refer to Paracharla's more detailed description of how to implement such queues.

159. At a high level, combining the concepts described by Petersen and Paracharla represents nothing more than a simple substitution of one known element (Petersen's virtual output queues) for another (Paracharla's flexible virtual output queues) to obtain predictable results (allowing the per-flow queueing of packets in a memory-efficient manner).

160. Additionally, combining the concepts described by Petersen and Paracharla is merely the use of a known technique (Paracharla's queueing of packets using per-flow virtual output queues) to improve a similar device (Petersen's network switching system) in the same way that per-flow queueing improved

Palacharla's switch, namely, by allowing for non-blocking data flows and reduced memory consumption.

D. Detailed Analysis of Claims

1. Claim 1

- a. [1.0] *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:*

161. To the extent that the preamble is limiting, Petersen discloses it.

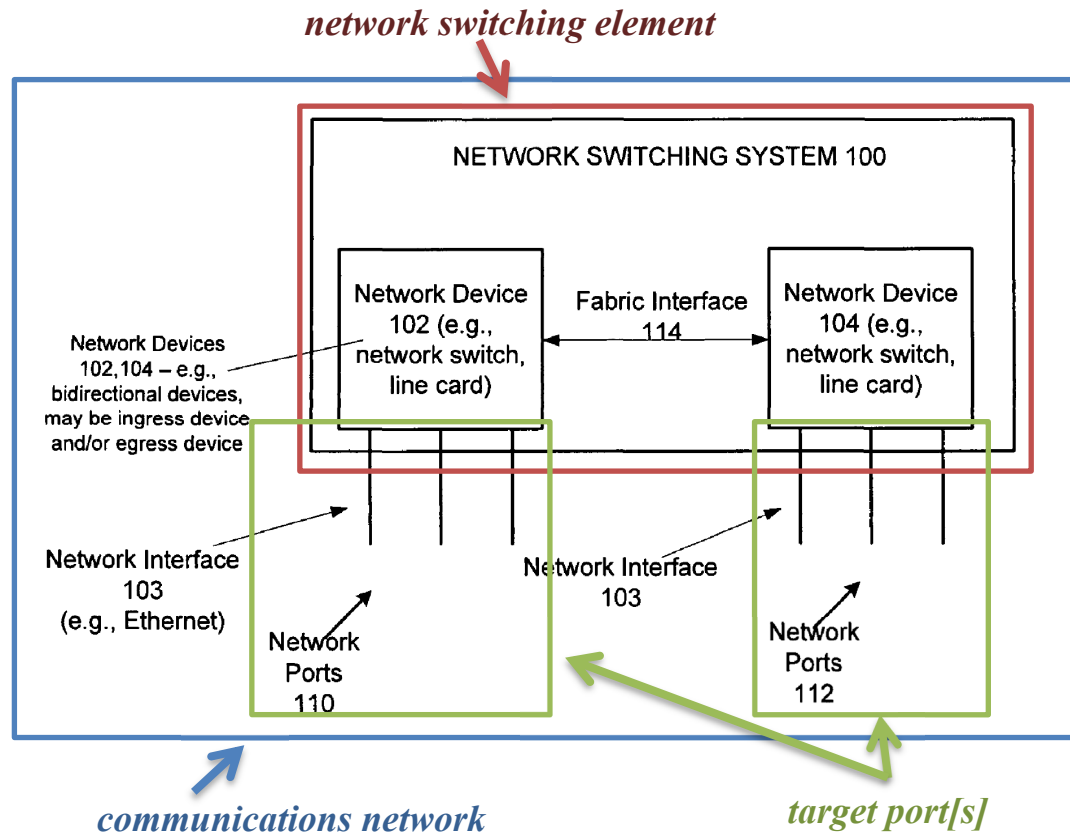
162. Petersen describes a “transport **method** that may be used for packet data transfer and related activity in a network switching system (or within a network switching environment.” Petersen at [0040]. Petersen’s method includes “making an **adjustment to data transmission rates.**” Petersen at [0082]. These adjustments are made so “that a traffic capacity (such as bit rate) of the shared resource is not exceeded by the total offered traffic load across the shared resource.” Petersen at [0092]. Petersen also explains that “traffic capacity” refers to a “bandwidth or bit rate capacity.” Petersen at [0093]. Petersen’s claim 18 makes this even more explicit: “**adjusting a transmission rate** from each the VOQs **to provide** an uneven or weighted **allocation of bandwidth** among the VOQs.” Petersen, claim 18. Petersen also describes “allocation algorithms” for fairly sharing bandwidth. Ex.1008, [0109]-[0110]. By adjusting transmission rates to provide an allocation of bandwidth, Petersen discloses a “*method of adjusting bandwidth allocation.*”

163. Petersen’s bandwidth adjustments are made in the context of a “network switching system,” such as the example system that Petersen illustrates in Fig. 1, below. Petersen at [0024]. The network switching system includes network devices providing network ports, such as switches, routers, or line cards:

Network switching system 100 may include one or more network devices, such as network devices 102, 104, etc. Network devices 102, 104 may be network switches, routers, line cards, or other network devices. Each network device may be a bidirectional device, and may be considered an ingress (or input) device and/or an egress (or output) device, depending on the direction of travel of a particular packet, for example. **Each network device may include one or more network ports**, such as network ports 110 for network device 102 and network ports 112 for network device 104.

Petersen at [0024].

164. Petersen’s network switching system corresponds to the claimed “*network switching element*.”



Petersen, Fig. 1 (annotated).

165. Petersen shows in Figure 1 that the network ports 110, 112 extend out from the network switching system 100, but the devices connected to the far ends of those connections are not shown in the figure. It would have been obvious to a POSITA, therefore, that the network switching system is part of a larger network including other network devices, such as computers, switches, and/or routers. This broader network corresponds to the claimed “communications network.”

166. Petersen also explains that the network ports 110, 112 may be “bidirectional,” such that they serve as both ingress ports and egress ports. Petersen

at [0027]. Any of Petersen's ports acting as an egress port corresponds to the claimed "*target port.*"

167. In summary, Petersen's techniques for fairly sharing bandwidth among ports of a network switching system renders obvious a "*method of adjusting bandwidth allocation by a network switching element in a communications network, the network switching element including a target port.*"

b. [1.1] *monitoring, by the network switching element, a data flow traversing the target port of the network switching element;*

168. Petersen, alone or in combination with Palacharla, renders this limitation obvious.

169. As discussed at [1.0], an egress port of Petersen's network switching system corresponds to the claimed "*target port of the network switching element.*"

170. Petersen describes identifying packets in a flow and adding a "flow header" at an ingress port that is then used to help transport the packets across the network switching system:

[0047] FIG. 6 is a diagram illustrating several examples of a header set 600 that may be used by a network switching system according to an example embodiment....

[0049] ... A third fabric header set 606 illustrates an example header set that may be assembled when two network switches or two network processors are exchanging packets within the network switching system,

e.g., across one or more intermediary devices. The header set 606 may include, for example, a **flow field identifying a flow (or group of associated fabric packets)**, a format field to identify a format for the fabric packet, and one or more processing parameters. The flow, format and processing parameter fields may, for example, be used by an egress device within the network switching system to perform the required processing during egress handling of the packet....

Petersen at [0047], [0049].

171. The flow header marks packets that are related or associated with one another in some way, such as by being directed to the same destination:

[0078] A flow header may also be used to identify a flow for the current packet. **A flow may be, for example, a group of related packets, or one or more packets from a same source or application, and/or directed to a same destination.** For example, a flow may be a source, destination and priority combination that defines a connection or aggregate of connections and/or that may originate from and/or are destined to a same or collocated network locations. **Other types of flows may be used as well.** The flow header may include a next header field to identify a next header after the current fabric header.

Petersen at [0078].

172. The destination may be an egress port of the network switching system:

A destination may include, for example, within a network switching system, an egress device, a port or physical port (e.g., **egress port**) of an egress device, a virtual port of an egress device, etc.

Ex.1008, [0036].

173. Petersen's "group of related packets" that are identified as a flow and are directed to the same egress port renders obvious the claimed "*data flow traversing the target port of the network switching element.*"

174. After Petersen's ingress port adds a flow header to a received packet, the ingress port places the packet in a virtual output queue (VOQ), which may be associated with a particular destination, such as a particular egress port:

Network device 500 may also include a plurality of **queues to store packets and/or cells to be transmitted or forwarded across a network switching system.** According to an example embodiment, virtual output queues (VOQs) may be used, such as VOQ1, VOQ2, VOQ3, ... VOQN for N VOQs. There may be, for example, **one virtual output queue (VOQ) at an ingress network device storing cells or packets to be transmitted to each destination.** Thus, within a network device 500 (as an ingress device), there may be an association between a virtual output queue (VOQ) and a destination, in an example embodiment. In an example embodiment, there may be one VOQ in an ingress device

per destination, for example. **A destination may include,** for example, within a network switching system, an egress device, a port or physical port (e.g., **egress port**) of an egress device, a virtual port of an egress device, etc.

Petersen at [0036].

175. The ingress port then monitors the status of its queues and reports their statuses—such as how full the queues are—to the other ingress ports or other components of the network switching system:

[0083] FIG. 8 is a diagram illustrating three different types of bandwidth management messages. Use of each of these sub-types of bandwidth management headers may result in the generation of a different bandwidth management message....

[0084] **The queue state message 810 may be used by ingress devices to convey or indicate the state of one or more VOQs** (virtual output queues) at the ingress device....

[0086] **The queue state message 810 may be transmitted** (e.g., to other devices in the network switching system) when there is a change in queue state (e.g., empty to non-empty, or increases in fullness), or may be sent out at specific time periods or intervals regardless whether there has been a change in queue state. Or, both techniques may be used to send out queue state messages (e.g., both when a change in a queue state occurs and

sending out a queue state message at specific times or intervals regardless of a change).

Petersen at [0083], [0084], [0086].

176. In summary, Petersen describes identifying related packets that are to be transmitted to a same destination (e.g., an egress port), marks them as relating to a “flow,” places the packets together in a queue, and then monitors the fullness of that queue. These steps and features collectively render obvious the claimed step of *“monitoring, by the network switching element, a data flow traversing the target port.”*

177. I understand that the Patent Owner may argue that the term “*data flow*” has a more specific meaning in the context of the ’884 patent. For example, the ’884 patent describes handling “data flows” in the context of Data Center TCP. ’884 patent, 1:33-41. From there, Patent Owner may argue that Petersen’s virtual output queue for a particular egress port would hold packets for multiple “*data flows*” (as Patent Owner defines that term) simply because all the packets are heading to the same egress port. Even under such an argument, however, I believe that the claim overall would have been obvious because enqueueing packets on a per-flow basis was known in the art. Specifically, Palacharla describes memory management techniques for providing “per-flow queues” in a scalable and efficient manner:

To accomplish non-blocking operation in a switch, **many switches incorporate** a large number of virtual output

queues (VOQs) for each non-blocking flow supported by the switching fabric. Such virtual output queues eliminate head of-line blocking by **queuing packets in per-flow queues** (i.e., separate queues for each combination of non-blocking input port, output port, and service level).

Palacharla, [0004]; *see also id.*, [0005]-[0008].

178. It would have been obvious to modify Petersen to implement per-flow virtual output queues, as taught by Palacharla. A POSITA would have been motivated to make such a combination because, as Palacharla acknowledges, per-flow queueing was commonly used by “many switches.” Palacharla, [0004]. And a POSITA would have had an expectation of success both because of that commonality, and because Petersen describes its queueing configuration as being flexible. For example, Petersen states that there may be “multiple VOQs” transmitting packets to the same destination, such as when a queue has “a traffic priority associated with it.” Petersen at [0037], [0110], [0033] (“merely some example system configurations”). A POSITA also would have expected success because Petersen describes a similarly flexible approach to grouping packets into flows. Petersen at [0078] (“Other types of flows may be used as well.”).

179. In summary, the combination of Petersen and Palacharla describe grouping packets into flows, placing related packets in a queue specific to each flow, and monitoring the status of the queue, which renders obvious “*monitoring, by the*

network switching element, a data flow traversing the target port of the network switching element.”

- c. **[1.2] *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;***

180. Petersen describes how each ingress port adjusts its transmission of packets toward an egress port based on the available bandwidth and the queue status reports of other ingress ports:

[0022] According to an example embodiment, each ingress device (or network device) may determine an overall or global traffic load offered for one or more destinations, path segments, or resources in a network switching system, e.g., based on received queue state messages received from the other ingress devices in the network switching system. The queue state messages may indicate, for example, a queue state (e.g., an empty/non-empty state or a level of fullness of the queue. A resource (e.g., path segment, intermediary device, or switch fabric device) may be shared among multiple active (e.g., non-empty) queues. The queues may be, for example, virtual output queues (VOQs). Based on a total offered traffic load for a shared resource, e.g., including traffic from multiple VOQs, one or more network devices in the network switching system may **adjust a transmission rate** from one or more of its active queues or VOQs over

the shared resource such that the total traffic load from a plurality of queues does not exceed the bandwidth or traffic capacity (e.g., bit rate capacity) of the shared resource.

Petersen at [0022].

181. As discussed further at [1.3] below, Petersen explains that this may be a “fair” sharing of the available bandwidth. Petersen at [0023]. However, Petersen also contemplates a situations where a single queue (e.g., a high priority queue) is entitled to use all of the available bandwidth:

The ingress device may allocate more (or even all) of the resource (e.g., larger percentage of available bit rate of shared segment) to the high priority VOQ. This may also be performed between ingress devices, where queue state messages indicate queue state as well as priority information of queue traffic. **A VOQ having higher priority traffic may be allocated a higher portion of the capacity or bit rate of the shared segment or resource.** Therefore, according to an example embodiment, a disproportionate amount of the traffic capacity of a resource may be allocated or available to high priority traffic, as compared to lower priority traffic. For example, all high priority VOQs (e.g., across multiple ingress devices) may share fairly the available capacity of the shared resource. **If there is only one high priority VOQ, that VOQ may claim or use all of the capacity or a**

higher percentage of the capacity than the lower priority traffic VOQs, for example. These are just a few additional examples of how capacity may be shared or allocated among multiple traffic sources (e.g., VOQs).

Ex.1008, [0110].

182. When one high-priority queue is using all the bandwidth capacity, it would have been obvious to a POSITA that the associated ingress port would have determined that the queue is “*currently allocated*” all available bandwidth. Since the ingress port is part of Petersen’s network switching system (see above, [1.0]), the determination of the “*bandwidth allocation*” would have been performed by the claimed “*network switching element*.” And since Petersen describes how each queue is associated with an egress port (see above, [1.1]), allocating all of the bandwidth capacity from the ingress port toward the egress port is making a “*bandwidth allocation for the target port*.”

183. The “*current*”-ness of the bandwidth allocation would have also been obvious from Petersen’s discussion of how the ingress ports regularly exchange queue state information. For example, queue fullness information would be exchanged among the ports periodically, whenever “a change in a queue state occurs,” or both. Petersen at [0086]. Petersen describes, in conjunction with Fig. 10, re-calculating bandwidth allocations upon receipt of updated queue information from other ports:

[0118] FIG. 10 is a flow chart illustrating operation according to an example embodiment. At 1010, a total offered traffic load may be determined for a shared resource within a network switching system, the total offered traffic load including a sum of offered traffic loads from one or more active virtual output queues (VOQs) of the network switching system.

[0119] For example, **operation 1010 may include each of a plurality of ingress devices of the network switching system reporting a state of one or more of its VOQs** such that one or more ingress devices of the network switching system obtains a global view of a demand (or offered traffic load) upon one or more shared resources of the network switching system. For example, queue state messages (e.g., indicating empty/non-empty state or a level of fullness) may be exchanged between network devices of the network switching system to allow one or more network devices of the system to determine a state (e.g., active or non-empty) of one or more VOQs. Each of the VOQs may be associated with a destination, and a path from an ingress device to an egress device may include one or more resources that may be shared among VOQs of the network switching system.

* * *

[0121] **At 1030, a transmission rate from one or more of the active VOQs over the shared resource may be**

adjusted such that the total traffic load from the active VOQs does not exceed the capacity of the shared resource. For example, operation 1030 may include adjusting a transmission rate from each (or one or more) of the VOQs that use the shared resource to be approximately a fair share (e.g., 1/Nth) of the capacity of the shared resource based on a number of active VOQs offering traffic load over the shared resource and the capacity of the shared resource.

Petersen at [0118]-[0121].

184. Thus, Petersen describes how ports would have regularly updated one another about their queue states and then updated their bandwidth allocations. It would have been obvious to a POSITA, therefore, for the ports to be continuously operating and transmitting packets in accordance with bandwidth that was “*currently allocated*” to each queue.

185. In summary, Petersen describes repeatedly updating bandwidth allocations for an ingress port’s queue of packets to be transmitted to an egress port, which renders obvious “*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.*”

186. Alternatively, Petersen and Palacharla also render this limitation obvious. As I discussed above at [1.2], the combination of Petersen and Palacharla

renders obvious employing per-flow queues in Petersen's ingress ports. Where there is only one high-priority queue with packets to transmit, it would have been obvious for that queue to be allocated all the available bandwidth, as Petersen describes. Petersen at [0110]. Since that queue would be associated with a single data flow (per Palacharla, [0004]), it would have been obvious that the bandwidth allocation would be "*bandwidth that is currently allocated for the data flow*" as claimed.

d. [1.3] *determining, by the network switching element, a fair-share bandwidth allocation for the target port,*

187. Petersen describes fairly sharing available bandwidth among multiple queues:

Based on this global demand of a shared resource, each network device may, for example, adjust a transmission rate for one or more of its active VOQs that are using the resource to use up to, for example, approximately **a fair portion (or other amount) of the bandwidth** or traffic (e.g., bit rate) capacity, such as approximately $1/N^{\text{th}}$ of the bandwidth or traffic capacity for the shared resource for N VOQs sharing the resource, according to an example embodiment.

Petersen at [0023].

For example, the ingress devices may **adjust the transmission rate for each VOQ to a fair portion of the capacity of the shared resource**, or $1/N^{\text{th}}$ of the capacity, where N maybe the total number of VOQs sharing the

resource (N=2 in this example). In this example, N=2, and the capacity of the path segment is 18 Gbps. Therefore, ingress devices 1 and 2 may adjust the traffic rate from VOQs 902 and 904, respectively, to 9 Gbps each, as shown in FIG. 9A.

Petersen at [0097].

For example, operation 1030 may include adjusting a transmission rate from each (or one or more) of the VOQs that use the shared resource to be approximately **a fair share (e.g., 1/Nth) of the capacity** of the shared resource based on a number of active VOQs offering traffic load over the shared resource and the capacity of the shared resource.

Petersen at [0121].

188. Petersen's description of a "fair share... of the capacity" for transmitting packets from an ingress port to an egress port renders obvious the claimed "*fair-share bandwidth allocation for the target port.*" Since Petersen describes adjusting the ingress port's actual transmission rate to match that fair-share capacity, it would have been obvious to a POSITA for Petersen's network switching system to "determin[e]" what that fair-share capacity is.

- e. [1.4] *the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and*

189. As I discussed at [1.4], Petersen renders obvious “*determining... a fair-share bandwidth allocation.*” Petersen explains that fair sharing is provided, for example, by dividing the available bandwidth by the number of queues sharing that bandwidth:

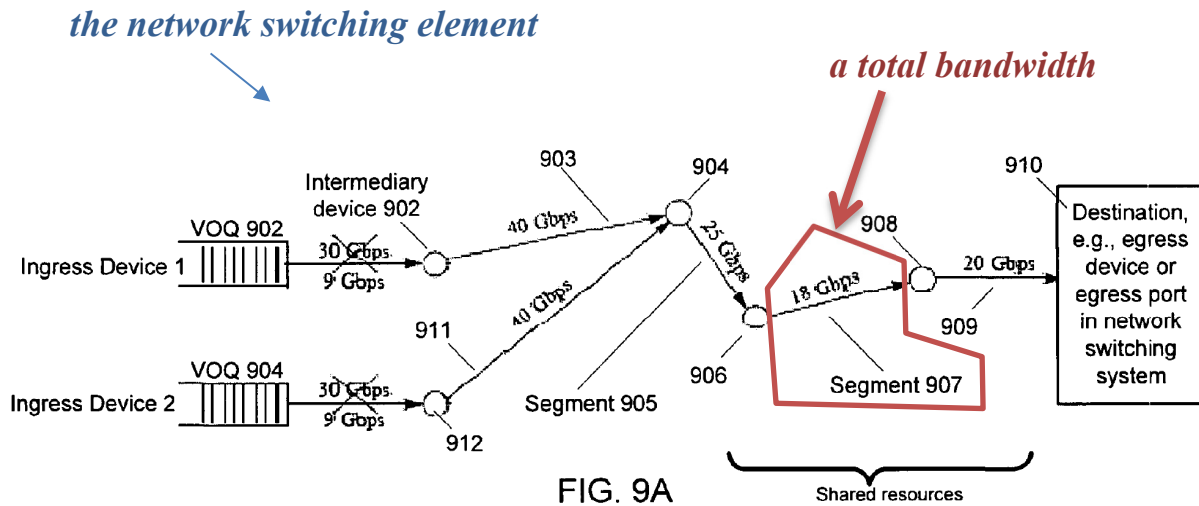
For example, the ingress devices may adjust the transmission rate for each VOQ to a fair portion of the capacity of the shared resource, or **1/Nth of the capacity, where N maybe the total number of VOQs sharing the resource** (N=2 in this example). In this example, N=2, and the capacity of the path segment is 18 Gbps. Therefore, ingress devices 1 and 2 may adjust the traffic rate from VOQs 902 and 904, respectively, to 9 Gbps each, as shown in FIG. 9A.

Ex.1008, [0097].

190. Petersen’s description of allocating “1/Nth of the capacity” is a “*proportional allocation.*”

191. The “shared resource” in Petersen corresponds to “*a total bandwidth of the network switching element.*” As discussed at [1.0], Petersen’s network switching system corresponds to the claimed “*network switching element.*” Petersen illustrates an example embodiment of the network switching system in Fig. 9A, below. In the

figure “segment 907 has a traffic capacity of 18 Gbps.” Petersen at [0096]. That bandwidth must be shared by two ingress devices (e.g., ingress ports) that both desire to transmit packets to the same egress port 910:



Petersen, Fig. 9A (annotated).

192. The available bandwidth on segment 907 is ultimately shared by the two ingress ports by dividing it equally (“a proportional allocation”), with each ingress port transmitting at 9 Gbps:

[0097] ... Thus, both ingress devices 1 and 2 may determine that there are two active VOQs which share the same resources, including path segment 907. Based on the offered traffic load of two active VOQs and a capacity of segment 907 of 18 Gbps, ingress devices 1 and 2 may adjust or decrease the transmission rate or traffic load from VOQs 902 and 904, respectively. For example, the ingress devices may adjust the transmission rate for each VOQ to

a fair portion of the capacity of the shared resource, or $1/N$ th of the capacity, where N maybe the total number of VOQs sharing the resource ($N=2$ in this example). In this example, $N=2$, and the capacity of the path segment is 18 Gbps. Therefore, ingress devices 1 and 2 may adjust the traffic rate from VOQs 902 and 904, respectively, to 9 Gbps each, as shown in FIG. 9A.

Petersen at [0097].

193. In summary, Petersen describes how ingress ports fairly share the available bandwidth to an egress port, which renders obvious “*the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element.*”

- a. [1.5] *adjusting, by the network switching element, the bandwidth allocation for the target port based on the fair-share bandwidth allocation.*

194. As discussed partly at [1.2]-[1.4] above, Peteresen describes how ingress ports adjust their transmission rates so as to fairly share the available bandwidth:

For example, the ingress devices may **adjust the transmission rate for each VOQ to a fair portion** of the capacity of the shared resource, or $1/N$ th of the capacity, where N maybe the total number of VOQs sharing the resource ($N=2$ in this example). Therefore, ingress devices

1 and 2 may adjust the traffic rate from VOQs 902 and 904, respectively, to 9 Gbps each, as shown in FIG. 9A.

Ex.1008, [0097].

Based on this global demand of a shared resource, each network device may, for example, **adjust a transmission rate** for one or more of its active VOQs that are using the resource to use up to, **for example, approximately a fair portion (or other amount) of the bandwidth** or traffic (e.g., bit rate) capacity, such as approximately $1/N^{\text{th}}$ of the bandwidth or traffic capacity for the shared resource for N VOQs sharing the resource, according to an example embodiment.

Ex.1008, [0023].

195. Continuing the specific example of Fig. 9A from [1.4] above, Petersen explains that the two ingress ports share the available 18 Gbps of bandwidth by each adjusting its transmission rate to 9 Gbps:

[0097] ... Based on the offered traffic load of two active VOQs and a capacity of segment 907 of 18 Gbps, ingress devices 1 and 2 may **adjust or decrease the transmission rate or traffic load** from VOQs 902 and 904, respectively. For example, the ingress devices may adjust the transmission rate for each VOQ to a fair portion of the capacity of the shared resource, or $1/N^{\text{th}}$ of the capacity, where N maybe the total number of VOQs sharing the resource (N=2 in this example). In this example, N=2, and

the capacity of the path segment is 18 Gbps. Therefore, ingress devices 1 and 2 may **adjust the traffic rate from VOQs 902 and 904, respectively, to 9 Gbps each**, as shown in FIG. 9A.

Petersen at [0097].

196. By adjusting ingress ports' transmission rates in accordance with fair sharing of the available bandwidth, Petersen renders obvious "*adjusting, by the network switching element, the bandwidth allocation for the target port based on the fair-share bandwidth allocation.*"

2. Claim 17

197. Claim 17 is substantially the same as claim 1, except that claim 17 is drafted as a device claim (directed to an "*edge switch*") instead of a method. Because claim 17 recites essentially the same subject matter as claim 1, claim 17 would have been obvious to a POSITA for the same reasons I provided above for claim 1. Even so, I will discuss below the specific, minor wording differences between the two claims.

- a. [17.0] *An edge switch for adjusting bandwidth allocation in a communications network, the edge switch including a target port, the edge switch configured to:***

198. To the extent that the preamble is limiting, it would have been obvious from Petersen.

199. As I discussed above for [1.0], Petersen describes a network switching system that would obviously be part of a larger “*communications network*” and that includes multiple ingress and egress ports (any of which correspond to “*a target port*”). A POSITA would have found it obvious to use Petersen’s network switching system as an “*edge switch*” because that was just one common use for network switches, and Petersen contemplates many “types or configurations of network switching systems.” Petersen at [0033]; *see also id.* [0003] (“Devices may be combined or coupled together in a variety of arrangements...”). For example, US 2013/0100858 describes a “well-known data center configuration” that includes a “first layer of switches” that are also referred to as “edge switch[es].” US 2013/0100858, [0023], Fig. 1.

200. As I discussed previously, it would have been obvious for Petersen’s network switching system to perform the steps of claim 1. For the same reasons, it would have been obvious for Petersen’s network switching system to be “*configured to*” perform the steps recited in claim 17.

b. [17.1] *monitor a data flow traversing the target port;*

201. See analysis at [1.1].

c. [17.2] *determine 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;*

202. See analysis at [1.2].

- d. [17.3] *determine a fair-share bandwidth allocation for the target port,*

203. See analysis at [1.3].

- e. [17.4] *the fair-share bandwidth allocation being a proportional allocation of a total bandwidth of the network switching element; and*

204. See analysis at [1.4]. I note that there is no earlier introduction of a “network switching element” in claim 17. For the purposes of my analysis, I assume that the “network switching element” recited here is supposed to refer to the “edge switch” introduced in the claim 17 preamble (limitation [1.0]). In my analysis of both [1.0] and [17.0], I map both these components to Petersen’s network switching system.

- f. [17.5] *adjust the bandwidth allocation for the target port based on the fair-share bandwidth allocation.*

205. See analysis at [1.5].

X. CONCLUSION

206. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and

Declaration of Paul Min, Ph.D.
Inter Partes Review of U.S. 9,450,884

the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: January 8, 2025



Paul Min, Ph.D.