

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

<i>In re</i> patent of Rafi Ram <i>et al.</i>	§	Attorney Docket No.: U022-0012RE
	§	
U.S. Patent 8,014,394	§	
	§	
Issue Date: September 6, 2011	§	Customer No.: 29150
	§	
Filing Date: February 11, 2009	§	
	§	
For: HIGH-SPEED PROCESSING OF	§	
MULTICAST CONTENT REQUESTS	§	

**REQUEST FOR *EX PARTE* REEXAMINATION OF
U.S. PATENT 8,014,394**

Mail Stop “*Ex Parte* Reexam”
Attn: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

Pursuant to the provisions of 35 U.S.C. §§ 301-307, Unified Patents, LLC (“Requester”) hereby requests an *ex parte* reexamination of claims 1-18 (the “Challenged Claims”) of U.S. Patent 8,104,394 (the “’394 Patent, Ex. 1001), which issued on September 6, 2011, from U.S. Patent Application 12/369,011 (“the ’011 Application”), filed on February 11, 2009. The ’394 Patent is currently assigned to Portsmouth Network Corporation. (“Portsmouth” or “Patent Owner”). The assignment is recorded in the U.S. Patent and Trademark Office (“USPTO”) at Reel 064870/Frame 0925.

Requester submits that this Request presents prior art references and analyses that are noncumulative of the prior art that was before the Examiner during the original prosecution of the ’394 Patent and that the Challenged Claims are invalid over these references. Requester therefore requests that an order for reexamination and an Office Action rejecting claims 1-18 be issued.

Ex Parte Patent Reexamination Filing Requirements

Pursuant to 37 C.F.R. § 1.510(b)(1), statements pointing out at least one substantial new question of patentability based on material, non-cumulative reference patents and printed publications for the Challenged Claims of the '394 Patent are provided in Section II of this Request.

Pursuant to 37 C.F.R. § 1.510(b)(2), reexamination of the Challenged Claims of the '394 Patent is requested, and a detailed explanation of the pertinence and manner of applying the cited references to the Challenged Claims is provided in Section III of this Request.

Pursuant to 37 C.F.R. § 1.510(b)(3), copies of every patent or printed publication relied upon or referred to in the statement pointing out each substantial new question of patentability or in the detailed explanation of the pertinence and manner of applying the cited references are provided as Exhibits 1001-1011 of this Request.

Pursuant to 37 C.F.R. § 1.510(b)(4), a copy of the '394 Patent is provided as Exhibit 1001 of this Request, along with a copy of any disclaimer, certificate of correction, and reexamination certificate issued corresponding to the patent.

Pursuant to 37 C.F.R. § 1.510(b)(5), the attached Certificate of Service indicates that a copy of this Request, in its entirety, has been served on Patent Owner at the following address of record for Patent Owner, in accordance with 37 C.F.R. § 1.33(c), as well as on Patent Owner's litigation counsel as a courtesy:

Patent Owner
May Patents Ltd.
c/o Dorit Shem-Tov
P.O.B 7230
Ramat-Gan 5217102
ISRAEL

Litigation Counsel
James R. Nuttall
Katherine H. Tellez
Robert F. Kappers
Daniel F. Gelwicks
STEPTOE & JOHNSON LLP
227 West Monroe Street, Suite 4700
Chicago, IL 60606

Also submitted herewith is the fee set forth in 37 C.F.R. § 1.20(c)(2).

Pursuant to 37 C.F.R. § 1.510(b)(6), Requester hereby certifies that the statutory estoppel provisions of 35 U.S.C. § 315(e)(1) and 35 U.S.C. § 325(e)(1) do not prohibit Requester from filing this *ex parte* patent reexamination request.

TABLE OF CONTENTS

Table of Exhibits.....	iii
I. U.S. Patent 8,014,394	1
A. Summary	1
B. Challenged Claims	4
1. “Upper-Limit” Claims	5
2. “Processor-Initializing” Claims	8
C. Prosecution History	11
D. Level of Ordinary Skill in the Art.....	15
E. Claim Construction	15
II. Listing of Prior Art Patents and Printed Publications.....	16
A. Naik (Ex. 1005).....	16
B. Reader (Ex. 1008)	21
C. Deng (Ex. 1007).....	24
D. Hu (Ex. 1009).....	25
E. Ramanathan (Ex. 1010)	27
F. Schulter (Ex. 1011)	30
III. Substantial New Questions of Patentability.....	32
IV. Detailed Application of the Prior Art to Every Claim for Which Reexamination Is Requested.....	35
A. Upper-Limit Claims	35
1. Ground 1: Naik in View of Reader Further in View of Hu Would Have Rendered Obvious Claims 1-7, 11-15, and 17	35
2. Ground 2: Naik in View of Reader Further in View of Hu, Further in View of Deng, Would Have Rendered Obvious Claim 16.....	95

B.	Processor-Initializing Claims.....	99
1.	Ground 3: Naik in View of Reader Further in View of Ramanathan and Schuler Would Have Rendered Obvious Claims 8, 9, and 18	99
2.	Ground 4: Naik in View of Reader, Ramanathan, and Schuler, Further in View of Deng Would Have Rendered Obvious Dependent Claim 10	117
C.	Secondary Considerations.....	118
V.	Disclosure of Concurrent Litigation, Reexamination, and Related Proceedings.....	118
VI.	Conclusion	118

TABLE OF EXHIBITS

<u>Exhibit</u>	<u>Description</u>
Ex. 1001	U.S. Patent No. 8,014,394 to Rafi Ram et al. ('394 Patent)
Ex. 1002	Prosecution History of the '394 Patent, available at https://patentcenter.uspto.gov/applications/12369011/ifw/docs .
Ex. 1003	Declaration of Anthony Wechselberger
Ex. 1004	<i>Curriculum Vitae</i> of Anthony Wechselberger
Ex. 1005	U.S. Patent App. Pub. No. 2006/0146857 (Naik)
Ex. 1006	U.S. Patent App. Pub. No. 2002/0191628 (Liu)
Ex. 1007	U.S. Patent App. Pub. No. 2005/0232272 (Deng)
Ex. 1008	U.S. Patent App. No. 2005/008901 (Reader)
Ex. 1009	U.S. Patent No. 8,068,416 (Hu)
Ex. 1010	U.S. Patent App. Pub. No. 2009/0279549 (Ramanathan)
Ex. 1011	U.S. Patent No. 7,174,390 (Schulter)

I. U.S. Patent 8,014,394

Before describing the substantial new questions of patentability, this Request provides an overview of the '394 Patent, lists the Challenged Claims, summarizes the prosecution history of the '394 Patent, and addresses the level of ordinary skill in the art and claim construction.

A. Summary

The '394 Patent is generally directed to “delivering multicast traffic over communication networks.” Ex. 1001 at 1:6-8. In these situations, a content source, such as an Internet Protocol Television (IPTV) distributor, may distribute media content to multiple clients, such as set-top boxes or other individual devices. *See id.* at 5:64-6:3, 6:15-22. The media content is communicated as streams of multicast packets addressed to multiple recipients. *Id.* at 5:61-64. Routers and other network equipment downstream from the content source help distribute the multicast traffic to the appropriate clients. Various standards and protocols govern the distribution of multicast traffic in a communication network, such as the Internet Group Management Protocol (IGMP), which is incorporated by reference in the '394 Patent.

According to the '394 Patent, the speed at which network equipment changes a routing or forwarding configuration for multicast traffic can affect certain client applications. *Id.* at 4:44-45. With IPTV, a need to change the forwarding configuration arises, for example, when a user switches (“zaps”) from one television channel to another. *Id.* at 4:46-50. The time after switching for a new IPTV channel to be displayed (“zapping time”) is driven by the time needed by the network equipment to change the forwarding configuration. *Id.* at 4:50-54. When the rate of change requests is high, “such as during a commercial break in which a large number of users switch channels almost simultaneously,” zapping time can be slowed. *Id.* at 4:54-57.

Purportedly to address these delays and other issues with multicast traffic, the '394 Patent discloses a method performed by a network element within a communication network. FIG. 1 in the patent, reproduced below and annotated outside a dashed frame, is a block diagram of the communication network for multicast traffic:

designated ports only if the stream appears in its filtering database. Otherwise, the NP discards the packets associated with the stream.” *Id.* at 7:18-23.

Network element 32 may update its filtering or forwarding database in response to clients sending to the element a multicast request, i.e., “a request to join a certain multicast service (i.e., to start receiving certain multicast content) or to leave a certain multicast service (i.e., to stop receiving certain multicast content).” *Id.* at 7:40-43. “When using IGMP, for example, the request may comprise an IGMP membership report message or an IGMP leave message.” *Id.* at 7:44-46. Processor 48 monitors incoming multicast requests and “processes the packets communicated over the ports assigned to the line card.” *Id.* at 6:32-34. When requests “to start or stop receiving certain multicast content” are detected, the “filtering databases in the different line cards are updated.” *Id.* 7:24-26; *see also id.* at 8:9-24. In permitting forwarding in response to a multicast request, the network element permits the requested stream to be forwarded over the port on which the request was received. *Id.* at 8:12-18 (“[I]f the request arrived over a certain ingress port, then NPs conclude that the multicast client that sent the request is reachable via this port. Therefore, the NPs update their filtering databases to permit forwarding of the requested multicast stream to this specific port.”).

Regarding the delay in changing a forwarding configuration (i.e., “zapping time”), the ’394 Patent describes a process of distributing new content requests to all processors in the network. Thus, if a given processing unit identifies a new multicast content request, that processing unit distributes the request “among the processing units in the network element.” *Id.* at 5:18-21, 7:56-61, FIG. 2 at 72. Then, “[t]he different processing units update their local filtering databases based on the distributed copy.” *Id.* at 5:23-25, FIG. 2 at Step 76. “Typically, the distributed copy of the request further comprises additional information needed by the NPs in order to process the request ... i.e., the port via which the request arrived at the network element.” *Id.* at 7:67-8:5. According to the ’394 Patent, following this process, multicast forwarding decisions for a given port “are carried out rapidly and locally by the processing unit assigned to the port, using its filtering database,” while the distributed configuration of the network element enables “the network element to update its multicast forwarding configuration with minimal latency.” *Id.* at 5:26-32.

Additionally, the ’394 Patent addresses a process for allowing “only a limited number of multicast streams to be forwarded over any given port, so as not to exceed the maximum bandwidths of the ports.” *Id.* at 8:36-39. Each port may be assigned a maximum allowed number

of multicast streams. In one example, a filtering database maintains “the number of additional streams that the port can still support” as a port counter. *Id.* at 8:44-45. Initially, the port counter equals the maximum allowed number of streams for the port. The processor decrements the counter as more streams are forwarded to the port and increments the counter as streams are removed from the port. *Id.* at 8:51-56. When a request for a new multicast stream arrives, the associated processor “may query the port counter of the ingress port, which is stored in database 52 of the ingress card, and check whether the current number of multicast streams permitted for forwarding over the ingress port has reached the maximum allowed number for that port.” *Id.* at 8:63-67. If so, the request for a new multicast stream is denied. *Id.* at 9:1-4.

Limit counters may also be used when two or more ports 44 in the same or different line cards are aggregated to form a high-capacity port 56 or a Link Aggregation Group (LAG), as shown above in annotated FIG. 1. *Id.* at 6:61-64. The patent acknowledges that link aggregation is standardized in “IEEE 802.1AX” and incorporates the standard by reference. *Id.* at 6:64-7:1. Network element 32 also “maintains a counter that limits the number of multicast streams that are permitted for forwarding over the LAG.” *Id.* at 9:64-67. The LAG limit counter may be distributed to different processors in the network element that host the different LAG members. *Id.* at 10:1-12. “This mechanism allows the NPs to manage local copies of the LAG limit counter without a need for additional communication or synchronization among the LAG members.” *Id.* at 10:12-14.

Finally, the ’394 Patent addresses a process for updating the filtering or forwarding database in a new line card added to an active network element. Initially, the line card disables its local ports 44 and sends “a multicast status update request (e.g., an IGMP general query) ... through all the ports of the network element.” *Id.* at 9:43-48. When the multicast content clients respond to the status update request, the processor can update its database, then activate the ports of the line card, and start normal operation. *Id.* at 9:48-53.

B. Challenged Claims

Requester seeks reexamination of all 18 claims in the ’394 Patent. As explained in the discussion of the prosecution history in Section I.C below, the independent claims of the ’394 Patent were granted after either of two limitations were incorporated from allowable dependent claims. Those incorporations led to two distinct claim sets in the ’394 Patent: claims 1-7 and 11-17 (referred to in this Request as “the Upper-Limit Claims” and addressed in Grounds 1 and 2)

and claims 8-10 and 18 (referred to in this Request as “the Processor-Initializing Claims” and addressed in Grounds 3 and 4). For reference, these two sets of claims are reproduced below. To aid the analysis and more easily identify claim limitations, Requester has added numbers within square brackets to certain phrases within the claims.

1. “Upper-Limit” Claims

In general, independent claims 1 and 11 of the Upper-Limit Claims contain final clauses relating, among other things, to “setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port.” *See* elements [1.7] and [11.7] below:

1. A method for communication, comprising:

[1.1] operating a network element, [1.1.1] wherein the network element further comprises multiple interconnected processing units and multiple ports, [1.1.2] wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and [1.1.3] wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned;

[1.2] storing in each processing unit a respective list, [1.2.1] each list indicating all of one or more multicast packet streams that are permitted for forwarding by the network element, and [1.2.2] further indicating which of the one or more ports each multicast packet stream is permitted to be forwarded to;

[1.3] receiving from a client a request relating to reception of a given multicast packet stream;

[1.4] [1.4.1] distributing the request among the multiple processing units within the network element, and [1.4.2] updating the respective list in each processing unit responsively to the request; and

[1.5] selectively forwarding multicast packets associated with the given multicast packet stream by the processing units in accordance with the respective updated lists,

[1.6] wherein operating the network element comprises aggregating two or more of the ports to form a high-capacity port, and

[1.7] wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port,

[1.7.1] wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and [1.7.2] wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.

2. The method according to claim 1, wherein storing the list, distributing the request and selectively forwarding the multicast packets are performed by dedicated hardware logic.

3. The method according to claim 1, [3.1] wherein distributing the request comprises duplicating the request to produce first and second copies, forwarding the first copy to a source of the given multicast packet stream, and [3.2] distributing the second copy among the multiple processing units.

4. The method according to claim 1, [4.1] wherein distributing the request comprises distributing among the multiple processing units an identity of an ingress port over which the request was received at the network element, and [4.2] wherein updating the list comprises causing the list to indicate that the given multicast packet stream is permitted for forwarding over the ingress port.

5. The method according to claim 1, [5.1] wherein the ports forming the high-capacity port are assigned to at least two of the processing units, and [5.2] wherein updating the list comprises holding respective local copies of the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, [5.3] updating the local copies based on the distributed request, and [5.4] selectively updating the list in each of the at least two processing units based on the respective local copies.

6. The method according to claim 1, wherein the request specifies a requested action selected from a group of actions consisting of receiving the given multicast packet stream and stopping to receive the given multicast packet stream.

7. The method according to claim 1, wherein the request conforms to one of an Internet Group Management Protocol (IGMP) and a Multicast Listener Discovery (MLD) protocol.

11. A network element, comprising;

[11.1] multiple ports; and

[11.2] multiple interconnected processing units, [11.2.1] wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and [11.2.2] wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned, [11.2.3] wherein the processing units are configured to process packets that are communicated over respective subsets of one or more of the ports,

[11.3] to store respective lists, [11.3.1] each of which indicates all of one or more multicast packet streams that are permitted for forwarding by the network element and [11.3.2] further indicates which of the one or more ports multicast packet stream is permitted to be forwarded to,

[11.4] to receive from a client a request relating to reception of a given multicast packet stream, and

[11.5] [11.5.1] to distribute the request among the processing units within the network element [11.5.2] so as to update the lists responsively to the distributed request and [11.5.3] to selectively forward multicast packets associated with the given multicast packet stream in accordance with the updated lists,

[11.6] wherein two or more of the ports are aggregated to form a high-capacity port, and

[11.7] wherein an upper limit is set on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port, [11.7.1] wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and [11.7.2] wherein the processing units are configured to update the lists to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.

12. The network element according to claim 11, wherein the processing units comprise dedicated hardware logic for storing the lists, distributing the request and selectively forwarding the multicast packets.

13. The network element according to claim 11, [13.1] wherein a processing unit receiving the request is configured to duplicate the request to produce first and second copies, so as to forward the first copy to a source of the given multicast packet stream and [13.2] distribute the second copy among the multiple processing units.

14. The network element according to claim 11, [14.1] wherein a processing unit receiving the request is configured to distribute among the multiple processing units an identity of an ingress port over which the request was received, and [14.2] wherein the processing units are configured to cause the respective lists to indicate that the given multicast packet stream is permitted for forwarding over the ingress port.

15. The network element according to claim 11, [15.1] wherein the ports forming the high-capacity port are assigned to at least two of the processing units, and [15.2] wherein the at least two processing units are configured to hold respective local copies of the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, [15.3] to update the local copies based on the distributed request, and [15.4] to selectively update their respective lists based on the respective local copies.

16. The network element according to claim 11, [16.1] wherein the processing units are configured to track a use of a permission to forward a certain multicast packet stream over a certain port by monitoring requests relating to the certain multicast packet stream that are received at the network element, and [16.2] to remove the permission from the respective lists responsively to determining that the permission is aged.

17. The network element according to claim 11, wherein the request conforms to one of an Internet Group Management Protocol (IGMP) and a Multicast Listener Discovery (MLD) protocol.

2. “Processor-Initializing” Claims

In general, independent claims 8 and 18 of the Processor-Initializing Claims contain final clauses relating, among other things, to “initializing a new processing unit added to the network element.” *See* elements [8.6] and [18.6] below:

8. A method for communication, comprising:

[8.1] operating a network element, [8.1.1] wherein the network element further comprises multiple interconnected processing units and multiple ports, [8.1.2] wherein each processing unit is assigned to a different subset of the multiple ports of the

network element, and [8.1.3] wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned;

[8.2] storing in each processing unit a respective list, [8.2.1] each list indicating all of one or more multicast packet streams that are permitted for forwarding by the network element, and [8.2.2] further indicating which of the one or more ports each multicast packet stream is permitted to be forwarded to;

[8.3] receiving from a client a request relating to reception of a given multicast packet stream;

[8.4] [8.4.1] distributing the request among the multiple processing units within the network element, and [8.4.2] updating the respective list in each processing unit responsively to the request; and

[8.5] selectively forwarding multicast packets associated with the given multicast packet stream by the processing units in accordance with the respective updated lists,

[8.6] wherein operating the network element comprises initializing a new processing unit added to the network element by:

[8.6.1] deactivating the ports to which the new processing unit is assigned;

[8.6.2] while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request, [8.6.3] receiving multicast membership reports sent by multicast clients in response to the multicast status request, [8.6.4] constructing the list by the new processing unit based on the received multicast membership reports; and

[8.6.5] after constructing the list, activating the ports to which the new processing unit is assigned.

9. The method according to claim 8, wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over a given port only when a number of the multicast packet streams that are permitted for forwarding over the given port, including the given multicast packet stream, does not exceed a maximum value specified for the given port.

10. The method according to claim 8, [10.1] wherein updating the list comprises tracking a use of a permission to forward a certain multicast packet stream over a certain port by monitoring requests relating to the certain multicast packet stream that are received at the network element, and [10.2] removing the permission from the list responsively to determining that the permission is aged.

18. A network element, comprising;

[18.1] multiple ports; and

[18.2] multiple interconnected processing units, [18.2.1] wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and [18.2.2] wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned, [11.2.3] wherein the processing units are configured to process packets that are communicated over respective subsets of one or more of the ports,

[18.3] to store respective lists, [18.3.1] each of which indicates all of one or more multicast packet streams that are permitted for forwarding by the network element and [18.3.2] further indicates which of the one or more ports multicast packet stream is permitted to be forwarded to,

[18.4] to receive from a client a request relating to reception of a given multicast packet stream, and

[18.5] [18.5.1] to distribute the request among the processing units within the network element [18.5.2] so as to update the lists responsively to the distributed request and [18.5.3] to selectively forward multicast packets associated with the given multicast packet stream in accordance with the updated lists,

[18.6] wherein a new processing unit added to the network element is configured to initialize by:

[18.6.1] deactivating the ports to which the new processing unit is assigned;

[18.6.2] while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request, [18.6.3] receiving multicast membership reports sent by multicast clients in response to the multicast status request, [18.6.4] constructing the list by the new processing unit based on the received multicast membership reports; and

[18.6.5] after constructing the list, activating the ports to which the new processing unit is assigned.

C. Prosecution History

As stated above, the '394 Patent issued from the '011 Application, which was filed February 11, 2009. Following two Office Actions rejecting the claims over prior art, the Examiner allowed the application when:

- the Applicant amended two independent claims to recite, in part, “setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port” (*see* elements [1.7] and [11.7] in independent patent claims 1 and 11); and
- the Applicant amended two other independent claims to recite, in part, “initializing a new processing unit added to the network element” (*see* elements [8.6] and [18.6] in independent patent claims 8 and 18).

Despite the differences in scope, no Restriction Requirement was made. The following outlines this development.

When filed, the '011 Application contained 37 claims. Ex. 1002 at 30-39.¹ For instance, representative independent claim 1 originally recited the following method for a network element:

1. (Application) A method for communication, comprising:
 - operating a network element, which includes multiple ports and multiple interconnected processing units, such that each processing unit is assigned to process packets that are communicated over a respective set of one or more of the ports;
 - storing in each processing unit a respective list, which indicates one or more multicast packet streams that are permitted for forwarding by the network element, and further indicates a respective subset of the ports over which each of the multicast packet streams is permitted for forwarding;
 - receiving from a client a request relating to reception of a given multicast packet stream;

¹ For convenience in citation, Requester has added consecutive numbering to the pages in the prosecution history in Exhibit 1002.

distributing the request among the multiple processing units, and updating the respective list in each processing unit responsively to the request; and

selectively forwarding multicast packets associated with the given multi cast packet stream by the processing units in accordance with the respective updated lists.

Id. at 30.

In a first Office Action, the Examiner rejected claims 1-6, 11-24, and 29-37 for being unpatentable under 35 U.S.C. § 103(a) over U.S. Patent App. Pub. No. 2006/0146857 (“Naik,” Ex. 1005) in view of U.S. Patent App. Pub. No. 2002/0191628 (“Liu,” Ex. 1006). *Id.* at 108-26. Additionally, the Examiner rejected dependent claims 9 and 27 over for being unpatentable under 35 U.S.C. § 103(a) over Naik in view of Liu, further in view of U.S. Patent App. Pub. No. 2005/0232272 (“Deng,” Ex. 1007). *Id.* at 126-29. The Office Action indicated that dependent claims 7, 8, 10, 25, 26, and 28 would be allowable if rewritten in independent form. *Id.* at 129.

In response to the first Office Action, the Applicant attempted to traverse the rejections, contending that the claimed invention was directed to reducing latency in routing multicast packet streams through a network. Ex. 1002 at 172. The Applicant further attempted to distinguish the network element recited in claim 1 from the router in Naik, while making several clarifying claim amendments. *Id.* at 161, 173-74.

In a Final Office Action, the Examiner maintained the claim rejections and rebutted the Applicant’s traversal. *Id.* at 188-213. The Examiner provided detailed explanation why each of the Applicant’s points did not overcome the rejections over the applied references. *Id.* at 210-13. The Final Office Action again indicated that the subject matter of dependent claims 7, 8, 10, 25, 26, and 28 would be allowable if rewritten in independent form.

In an Amendment After Final, the Applicant acquiesced to the rejections and rewrote four of the allowable dependent claims in independent form. For example, the Applicant rewrote dependent claim 7 into independent form, resulting in the following limitations being included with the subject matter of independent method claim 1:

wherein operating the network element comprises aggregating two or more of the ports to form a high-capacity port

wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port, wherein the request

indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.

Id. at 219-20. Amendments containing essentially the same limitations were made in rewriting allowable apparatus claim 25. *Id.* at 223-24. As amended, proposed claims 7 and 25 became the two independent claims of the Upper-Limit Claims (patent claims 1 and 11).

In addition, the Applicant rewrote allowable claim 10 in independent form, which included the following limitations with the subject matter of independent method claim 1:

wherein operating the network element comprises initializing a new processing unit added to the network element by:

deactivating the ports to which the new processing unit is assigned;

while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request, receiving multicast membership reports sent by multicast clients in response to the multicast status request, constructing the list by the new processing unit based on the received multicast membership reports; and after constructing the list, activating the ports to which the new processing unit is assigned.

Id. at 221-22. Amendments containing essentially the same limitations were made in rewriting allowable apparatus claim 28 in independent form. *Id.* at 225-26. As amended, proposed claims 10 and 28 became the independent claims of the Processor-Initializing Claims (patent claims 8 and 18).

The Office mailed a Notice of Allowance on May 12, 2011. *Id.* at 232. The Examiner identified the amended subject matter as the Reasons for Allowance of each independent claim:

3. The following is an examiner's statement of reasons for allowance:

Regarding claim 7 and subsequent dependant claims, prior art of record fails to teach or render obvious, alone or in combination, the method for communication wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port, wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit, along with all other disclosed limitations as recited in independent claim 7.

Ex. 1002 at 238 (discussing proposed claim 7, issued as claim 1); *see also id.* at 238-39 (similar reasoning for proposed claim 10, issued as claim 8), 239 (similar reasoning for proposed claim 25, issued as claim 11), 239-40 (similar reasoning for proposed claim 28, issued as claim 18).

The '394 Patent issued September 6, 2011. Application claims 2-5, 7-12, 20-22, 25-28, and 30 issued as patent claims 2-4, 9, 1, 5, 10, 8, 6, 7, 12-14, 11, 15, 16, 18, and 17, respectively. *Id.* at 242.

Thus, the points of novelty for independent claim 1 are:

wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port, wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.

Ex 1001 at 12:21-32 (elements [1.7]). Claim 11 includes the same points of novelty phrased for an apparatus. *Id.* at 14:6-16 (elements [11.7]).

And the points of novelty for independent claim 8 are:

wherein operating the network element comprises initializing a new processing unit added to the network element by:

deactivating the ports to which the new processing unit is assigned;

while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request, receiving multicast membership reports sent by multicast clients in response to the multicast status request, constructing the list by the new processing unit based on the received multicast membership reports; and after constructing the list, activating the ports to which the new processing unit is assigned.

Id. at 13:21-33 (elements [8.6]). Claim 18 includes the same points of novelty phrased for an apparatus. *Id.* at 15:11-16:10 (elements [8.6]).

D. Level of Ordinary Skill in the Art

A person of ordinary skill in the art (“POSITA”) at the time of the ’394 Patent’s filing date of February 2009 would have had a bachelor’s degree in computer science or computer engineering and two years of experience in computer and networking technology. Ex. 1003 ¶ [0027]. Less work experience may be compensated by a higher level of education, such as a master’s degree, and vice versa. *Id.*

E. Claim Construction

Because the ’394 Patent has not expired, the Challenged Claims should be given their “broadest reasonable interpretation consistent with the specification.” 37 C.F.R. §1.75(d)(1); see also MPEP § 2111; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005). Under this standard, “[t]he Patent and Trademark Office ... determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction in light of the specification as it would be interpreted by one of ordinary skill in the art.” *Phillips*, 415 F.3d at 1316 (quotation and citation omitted). “Because applicant has the opportunity to amend the claims during prosecution, giving a claim its broadest reasonable interpretation will reduce the possibility that the claim, once issued, will be interpreted more broadly than is justified.” MPEP § 2111 (citing *In re Yamamoto*, 740 F.2d 1569, 1571 (Fed. Cir. 1984)).

Because the Challenged Claims are unpatentable under any reasonable construction, no construction is necessary for this reexamination.

II. Listing of Prior Art Patents and Printed Publications

As stated above, the '394 Patent issued from the '011 Application, which was filed February 11, 2009. Because the filing date for the '394 Patent was before March 16, 2013, when the America Invents Act became effective, the “pre-AIA” patent laws of Title 35 define the prior art for the '394 Patent. February 11, 2009, is the constructive date of invention for determining whether a document is prior art under 35 U.S.C. §§ 102(a) and (e) (pre-AIA), and February 11, 2008 is the critical date for determining prior art under 35 U.S.C. § 102(b) (pre-AIA).

Requester requests reexamination of the Challenged Claims in view of the following prior art documents and new technological teachings:

<u>Exhibit</u>	<u>Prior Art Documents</u>
Ex. 1005	U.S. Patent App. Pub. No. 2006/0146857 (Naik)
Ex. 1008	U.S. Patent App. Pub. No. 2005/0080901 (Reader)
Ex. 1007	U.S. Patent App. Pub. No. 2005/0232272 (Deng)
Ex. 1009	U.S. Patent No. 8,068,416 (Hu)
Ex. 1010	U.S. Patent App. Pub. No. 2009/0279549 (Ramanathan)
Ex. 1011	U.S. Patent No. 7,174,390 (Schulter)

Naik and Deng were cited and applied during prosecution of the '394 Patent, but none of the other references listed above was cited or considered during prosecution. A Form SB-08 and copies of the cited references are submitted with this Request. The declaration of Anthony Wechselberger (Ex. 1003) supports this Request.

The following summarizes the prior art patents and printed publications relied on in this Request and listed above.

A. Naik (Ex. 1005)

U.S. Patent App. Pub. No. 2006/0146857 (“Naik,” Ex. 1005) was filed December 30, 2004, and published July 6, 2006. Accordingly, Naik is prior art to the '394 Patent under 35 U.S.C. § 102(b) (pre-AIA). Naik was applied in two Office Actions during prosecution, in which the Examiner concluded that the reference discloses the subject matter of at least elements [1.1]-[1.3] in independent patent claim 1 and elements [11.1]-[11.3] in independent patent claim 11. Ex. 1002

at 188-91, 197-99. The Applicant did not successfully rebut these conclusions. *Id.* at 171-77, 210-213.

Naik is directed to a system and method for controlling admission of an end user to a multicast channel over a network. Naik includes an extensive overview of multicasting and Internet Group Management Protocol (IGMP). As Naik explains, multicasting refers to the transmission of data packets from a content source to “more than one, but less than all of the receivers on the network.” Ex. 1005 ¶ [0005]. The source specifies a multicast group address for the multicast, and an end user, or host, can join or leave the multicast by sending a multicast request upstream in the network to a multicast router. *Id.* ¶¶ [0006]-[0007]. Under different versions of IGMP, a group membership message to the router, often called a Membership Report, may include the multicast request. *Id.* ¶ [0063], FIGS. 6B-6C; Ex. 1003 ¶ [0056]. The router may also receive a Membership Report from users in response to sending a Membership Query to learn about the status of multicast membership. Ex. 1005 ¶ [0060], FIG. 6A; Ex. 1003 ¶ [0056].

Naik uses various names for a multicast request from a user. These include: a “membership report” (Ex. 1005 ¶ [0060]), a “group membership message” (*id.* ¶ [0007]), a “report message” (*id.* ¶ [0063]), an “IGMPV3 join report” (*id.* ¶ [0073], FIG. 7), an “IGMP join” (*id.* ¶ [0020]), “IGMP join and leave packets” (*id.* ¶ [0024]), “IGMPv3 packets” (*id.* ¶ [0054]), a “join message” (*id.* ¶ [0070], [0073]), a “join request” (*id.* ¶ [0073]), and “requests to ‘join’ a group or ‘leave’ a group” (*id.* ¶ [0057]). Ex. 1003 ¶ [0057].

In that context, Naik discloses multicast edge router R1 (FIG. 2) operating as an intermediate node 200 within a computer network 100 (FIG. 1). Ex. 1005 ¶¶ [0002]-[0022], [0036], [0040]. As shown in FIG. 2, reproduced below and annotated outside a dashed frame, edge router R1 includes route processors 210 interconnected with line cards 260 via switch fabric 250:

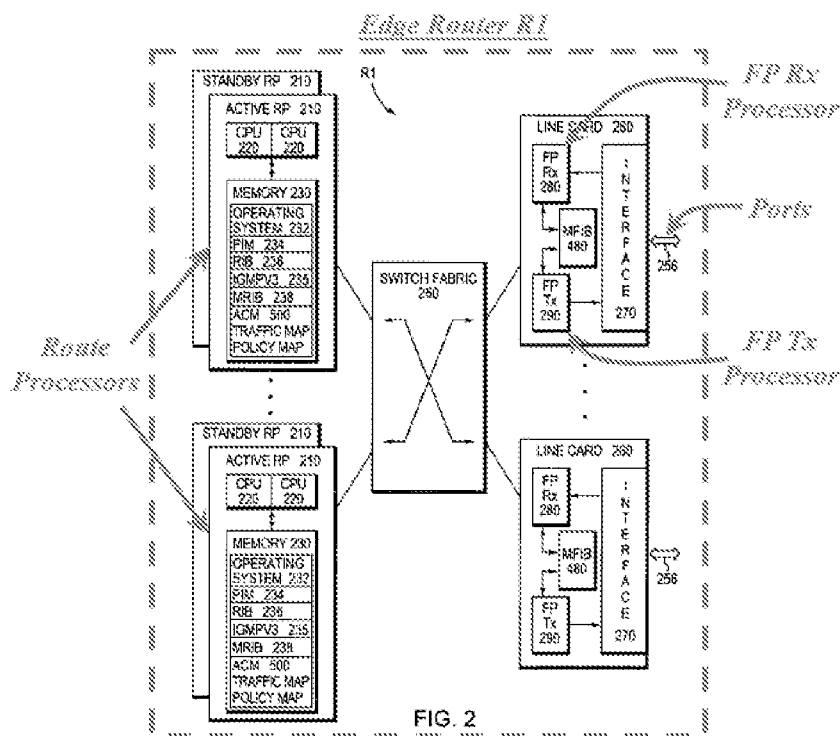


FIG. 2

Id. ¶ [0040]. For receiving and transmitting data, line cards 260 include ports 256 coupled to an interface 270, a receive forwarding processor FP Rx 280, and a transmit forwarding processor FP Tx 290. *Id.* ¶ [0046].

According to Naik, “the route processors 210 handle routing and control of the data traffic forwarding,” while “line cards 260 perform forwarding of the data traffic.” Ex. 1005 ¶ [0046]. To track routing information, route processors 210 execute a Multicast Route Information Base (MRIB) that “provides the routing information and a multicast-capable path to each multicast destination.” *Id.* ¶ [0045]. Executing in line cards 260, Multicast Forwarding Information Bases (MFIBs) are derived from the MRIB. The MFIB “is embodied as one or more multicast forwarding tables 400 that are populated on the line cards 260 and whose contents describe how to forward data packets through the router.” *Id.* Receive forwarding processor FP Rx 280 and transmit forwarding processor FP Tx 290 make use of MFIB 480 to determine where to forward each packet received at, or transmitted from, a port. *Id.* ¶ [0046]; *see also* Ex. 1003 ¶ [0060].

FIG. 4 of Naik, reproduced below, illustrates an MFIB forwarding table 400 with each row corresponding to an entry 410 with various characteristics for a packet:

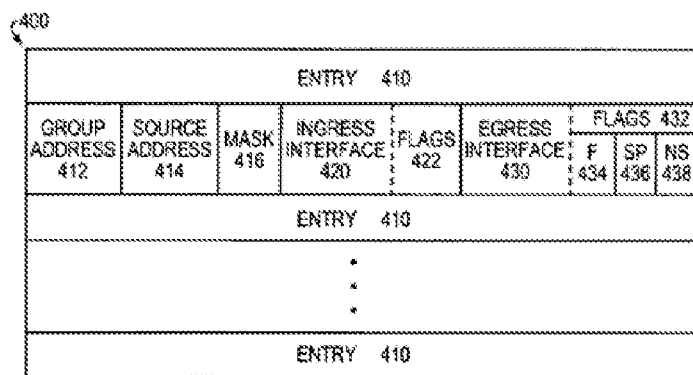


FIG. 4

As explained in Naik, each entry for a packet in the forwarding table may include, for example, a multicast group address field 412, a source address field 414, an ingress interface field 420 specifying “an interface on which an incoming multicast packet should be accepted,” an egress interface field 430 “over which the incoming packet should be forwarded,” and one or more control flags 432 specifying certain actions that may be taken by the router in response to receiving the incoming packet. Ex. 1005 ¶¶ [0050]-[0051]. “In response to receiving the incoming packet from a neighbor, the MFIB 480 performs a lookup into its forwarding table 400 to find a route of an entry 410 that matches a destination address of the packet. The matching route instructs the router as to which interfaces the packet should be forwarded.” *Id.* ¶ [0052].

An Admission Control Module (ACM) 500 executed by the route processors 210 in Naik controls admission of an end user to a multicast channel using a filtering protocol. *Id.* at Abstract. The ACM monitors an “incoming request issued by the end nodes to join, leave or modify membership in multicast groups designated to receive data.” *Id.* ¶ [0053]. Specifically, the ACM comprises a “snooping” process that examines incoming IGMP packets and extracts information, such as source address, multicast group address, and port numbers. *Id.* ¶ [0025]. The ACM snoops packets “to learn and maintain group memberships and determine which ports are requesting to receive which multicast addresses from which source.” *Id.* ¶ [0066]. The ACM then stores which ports are receiving multicast requests, so that when the multicast traffic is received, “it is directed only to [] those ports so interested. The traffic is not directed to those ports where no end user either requested or has the status to receive the multicast.” *Id.*; *see also* Ex. 1003 ¶ [0063].

After snooping the packets, ACM 500 applies the extracted information to parameters of a traffic map. If the information matches the parameters, ACM 500 enters a policy map 508 where

any reasons to deny admission or otherwise limit behavior are determined. *Id.* ¶¶ [0025], [0055]; *id.* ¶ [0023] (“The ACM accepts or denies these requests based on a configurable service policy.”). Accordingly, the IGMP snooping process of ACM 500 “cooperates with the traffic and policy maps to control access to the video channels at the hosts.” *Id.* If no reasons to deny are found, “[t]he actions include accepting membership in a group associated with a multicast channel and pushing the packets to the end user.” *Id.* at Abstract.

Naik explains that ACM 500 may deny admission to a multicast channel if a maximum number of channels allowed on a port has been reached. Ex. 1005 ¶¶ [0025], [0055], [0068]; *id.* at Abstract (“the actions may be to limit the number of routing states, by denying admittance to a groups [sic] once a limit number of requests to join, or other such parameter, is reached”). Limiting the number of channels or hosts joined to a channel on a port can help deter Denial of Service attacks:

DOS attacks try to create large streams of joins/leaves that will cause the PIM to create large routing structures or tie up the processors. If unchecked such attacks will eventually crash the system. The ACM, in an illustrative embodiment, would be loaded with a predetermining a maximum number of “joins” allowed by the switch, or “joins” received from a port, or the number of ports subscribed for a channel, or combinations thereof or of other relevant parameters. The ACM would track the number involved and compare it to the limit. When the limit is reached any additional joins will always be denied regardless of the traffic and service policy. The limit parameter may be stored in the tracking table described below, but in any event the PIM 234 and the MRIB 238 may participate in implementing such limits.

Id. ¶ [0072]; *see also id.* ¶¶ [0023], [0026], [0053].

Naik is analogous art to the '394 Patent. Ex. 1003 ¶ [0066]. “A reference is analogous art to the claimed invention if: (1) the reference is from the same field of endeavor as the claimed invention (even if it addresses a different problem); or (2) the reference is reasonably pertinent to the problem faced by the inventor (even if it is not in the same field of endeavor as the claimed invention).” MPEP § 2141.01(a) (*citing In re Bigio*, 381 F.3d 1320, 1325 (Fed. Cir. 2004)). The '394 Patent and Naik are in the same field of endeavor, namely, developing devices and methods for forwarding data packets in a communications network, particularly multicast packet streams. Ex. 1001 at 1:6-8; Ex 1005 ¶ [0001]. In addition, Naik is reasonably pertinent to a problem faced by the Upper-Limit Claims of the '394 Patent at least in that both documents seek to control access

to multicast packet streams at a network switch. Ex. 1001 at 8:36-45, 12:11-32; Ex. 1005 at Abstract. For example, the '394 Patent describes setting an upper limit on a number of multicast channels permitted on a switch port and denying a new multicast join request that would exceed the limit (*see* Ex. 1001 at 8:36-9:4), while Naik describes an edge router including an Admission Control Module that will deny a new multicast join request if a limit on the number of join requests is reached (*see* Ex. 1005 ¶¶ [0053], [0072]). At least by also seeking to control access to multicast packet streams at a network switch, Naik is analogous art to the '394 Patent. Ex. 1003 ¶ [0066].

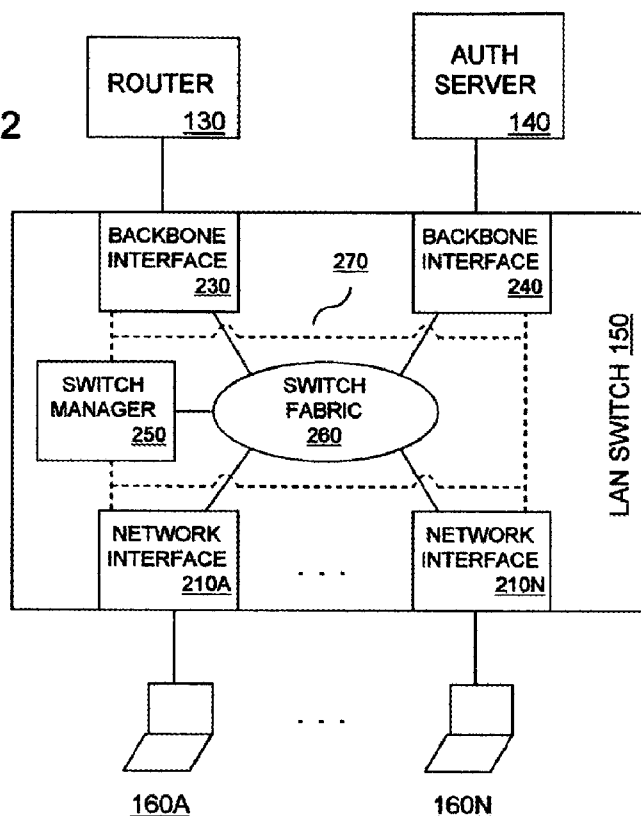
B. Reader (Ex. 1008)

U.S. Patent App. Pub. No. 2005/0080901 ("Reader," Ex. 1008) was filed October 14, 2003, and published April 14, 2005. Accordingly, Reader is prior art to the '394 Patent under 35 U.S.C. § 102(b) (pre-AIA). Reader was not cited or otherwise considered during prosecution of the '394 Patent. *See* Ex. 1001 at pp. 1-2; Ex. 1002 at 132, 145-51.

Reader discloses a method and apparatus for controlling access to multicast data streams within a data communication network. Ex. 1008 at Title, Abstract, FIGS. 1-2, ¶ [0001]; *see also* Ex. 1003 ¶ [0068]. According to Reader, conventional network switches for local area networks (LANs) "forward multicast traffic on switch ports without regard to whether the switch ports support an end station that is a destination host for the multicast." *Id.* ¶ [0003]. Therefore, while a conventional LAN switch "snoops" a received IGMP membership report to associate a multicast group with the ingress switch port, "the LAN switch does not render any threshold decision as to whether to allow the end station to receive traffic addressed to the group based on a network policy, such as machine or user identity." *Id.* ¶ [0004].

To enforce a network policy on multicast traffic, Reader discloses a LAN switch 150 coupled to an authentication server 140 within a communication network in FIG. 2:

Figure 2



LAN switch 150 includes network interfaces 210A-210N for communicating with end stations 160A-160N, respectively, and backbone interfaces 230, 240 for communicating with router 130 and authentication server 140, respectively. *Id.* ¶ [0018]. For wired links, “network interfaces 210A through 210N communicate with their respective end stations 160A through 160N via a dedicated physical port on network interfaces 210A through 210N.” *Id.* Interfaces 210A through 210N, 230, 240 may be implemented through various electronics, including “a combination of discrete logic and programmable logic, such as programmable network processors.” *Id.*

For forwarding IP Multicast data packets according to a network policy, a switch manager 250 within LAN switch 150 maintains a master multicast forwarding database (MM-FDB) 350 and a multicast authorization database (M-ADB) 340. *Id.* ¶¶ [0019], [0021], [0024], [0025], FIG. 3. The master forwarding database MM-FDB 350 “has as entries multicast groups and associated switch ports through which destination hosts that are registered in the multicast groups are reachable,” while the multicast authorization database M-ADB 340 has “as entries switch ports and associated multicast group addresses or address ranges for which the switch ports are authorized.” *Id.* ¶¶ [0021], [0025].

Interfaces 210A-210N, 230, and 240 snoop incoming packets and, if an IGMP membership report is identified, “the packet is transmitted to switch manager 250.” *Id.* ¶ [0025]. Switch manager 250 then determines from the multicast authorization database M-ADB 340 “whether the multicast group address specified in the report is within the permitted or proscribed multicast group addresses or address ranges specified for the switch port.” *Id.* If the port is authorized, switch manager 250 “updates MM-FDB 350 to include the new multicast group/port association, and relays the packet to router 130 via backbone interface 240.” *Id.* If not authorized, “the packet is dropped without updating MM-FDB 350.” *Id.*

Reader explains that in addition to switch manager 250 maintaining master multicast forwarding database (MM-FDB) 350, each of the interfaces 210A-210N, 230, and 240 maintains slave multicast forwarding databases (SM-FDBs). *Id.* ¶ [0021]. Upon receiving updates to master multicast forwarding database (MM-FDB) 350, switch manager distributes the updates to the slave forwarding databases:

Switch manager 250 distributes the contents of MM-FDB 350 to interfaces 210A through 210N, 230, 240 in response to updates for MM-FDB 350 and thereby maintains slave multicast forwarding databases (SM-FDBs) on interfaces 210A through 210N, 230, 240.

Id.; *see also id.* ¶ [0023] (“The contents of MU-FDB and MM-FDB 350 are distributed by switch manager 250 to interfaces 210A through 210N, 230, 240”). As a result, the slave forwarding database (SM-FDB) on one of interfaces 210A-210N, 230, or 240 where an IP Multicast data packet arrives can be “invoked to resolve a multicast group address” for the packet to one or more switch ports where it should be forwarded. *Id.* ¶ [0021]; Ex. 1003 ¶ [0072].

Reader is analogous art to the '394 Patent. Ex. 1003 ¶ [0073]. First, the '394 Patent and Reader are in the same field of endeavor, namely, developing devices and methods for forwarding data packets in a communications network, particularly multicast packet streams. Ex. 1001 at 1:6-8; Ex. 1008 ¶ [0001]; *see* MPEP § 2141.01(a). Second, Reader is reasonably pertinent to a problem faced by the Upper-Limit Claims of the '394 Patent at least in that both documents seek to control access to multicast packet streams at a network switch. Ex. 1001 at 8:36-45, 12:11-32; Ex. 1005 ¶ [0001]. For example, the '394 Patent describes setting an upper limit on a number of multicast channels permitted on a switch port and denying a new multicast join request that would exceed the limit (*see* Ex. 1001 at 8:36-9:4), while Reader describes a LAN switch within a communication network that includes an authentication server for checking whether a new multicast join request

is authorized by a network policy (*see* Ex. 1008 ¶ [0025]). At least by also seeking to control access to multicast packet streams at a network switch, Reader is analogous art to the '394 Patent. Ex. 1003 ¶ [0073].

C. Deng (Ex. 1007)

U.S. Patent App. Pub. No. 2005/0232272 (“Deng,” Ex. 1007) was filed June 15, 2005, and published October 20, 2005. Accordingly, Deng is prior art to the '394 Patent under 35 U.S.C. § 102(b) (pre-AIA). Deng was applied in two Office Actions during prosecution in the form of its child application, U.S. Patent App. Pub. No. 2009/0290585. The Examiner concluded that the reference discloses the subject matter of dependent application claims 9 and 27 (now patent claims 10 and 16). Ex. 1002 at 126-29, 207-09. The Applicant did not rebut these conclusions. *Id.* at 175, 177-78.

Deng discloses a network switch for a data link layer that includes a switching mechanism coupled to port interface controllers. Ex. 1007 at Abstract. Each of the port interface controllers contains a multicast address table, a multicast matcher, and a timer. When an incoming packet is a multicast join request and matches a target pattern, the multicast matcher generates a multicast address from the packet, stores the generated address in the multicast address table, and “starts or restarts the timer.” *Id.* ¶ [0036]. The timer “is used to determine the length of time that an address remains in the multicast address table 320. An address is removed from the table upon the expiration of the timer.” *Id.* Thus, “[t]he timer determines an amount of time that the generated multicast address remains in the multicast address table.” *Id.* at Abstract; *see also* Ex. 1003 ¶ [0075].

Deng is analogous to the subject matter of the Challenged Claims. Ex. 1003 ¶ [0076]. For example, the '394 Patent and Deng are from the same field of endeavor, namely, developing devices and methods for forwarding data packets in a communications network, particularly multicast packet streams. Ex. 1001 at 1:6-8; Ex 1007 ¶ [0002]. In addition, Deng is reasonably pertinent to a problem faced by the '394 Patent in the Challenged Claims at least in that both documents seek to control access to multicast packet streams at a network switch, such as by retaining multicast stream memberships that are refreshed periodically. Ex. 1001 at 9:24-40, 13:41-47; Ex. 1007 at Abstract. For example, the '394 Patent describes keeping track of membership for a particular switch port in a multicast stream using an aging counter. Ex. 1001 at 9:27-30. When old entries are not refreshed recently, they are removed. *Id.* at 9:35-37. Likewise, in Deng, a timer

310 is used to determine the length of time an address remains in a multicast address table 320, and the address is removed when the timer expires. Ex. 1007 ¶ [0036]. At least by also seeking to control access to multicast packet streams at a network switch, Deng is analogous art to the '394 Patent. Ex. 1003 ¶ [0076].

D. Hu (Ex. 1009)

U.S. Patent No. 8,068,416 ("Hu," Ex. 1009) issued November 29, 2011, from an application filed September 20, 2007, and is prior art to the '394 Patent under 35 U.S.C. §§ 102(a) and 102(e) (pre-AIA). Hu was not cited or otherwise considered during prosecution of the '394 Patent. *See* Ex. 1001 at pp. 1-2; Ex. 1002 at 132, 145-51.

Hu is directed to a method of allocating media streams on a network content switch of an Internet Protocol Television (IPTV) network or other packet-based video distribution network. Ex. 1009 at 2:56-61; *see also* Ex. 1003 ¶ [0078]. A content switch 102, as shown in FIG. 1 below, within system 100 has "processing logic 104, such as one or more independent or redundant processors," memory 106, and physical ports such as 122 and 126 each logically grouped into a "link aggregation group." *Id.* at 3:7-19.

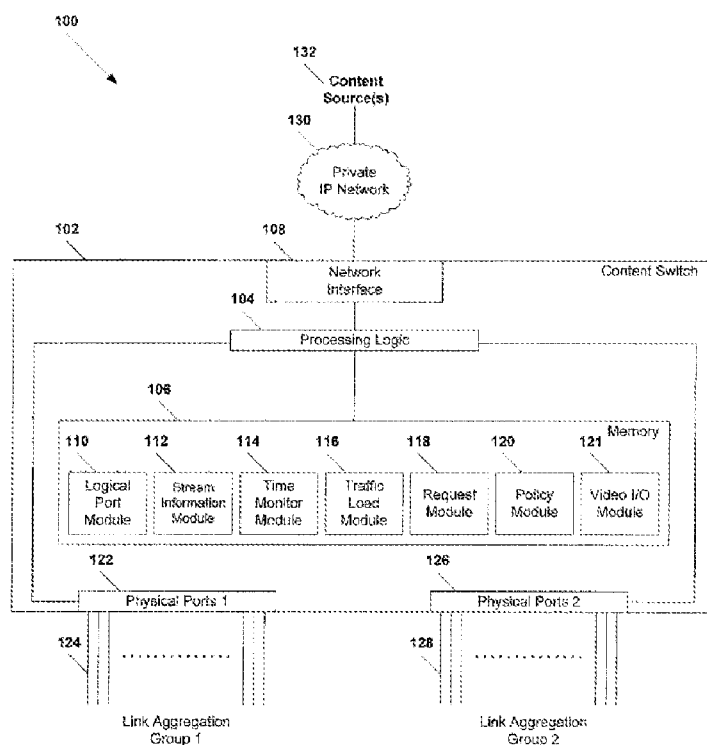


FIG. 1

According to Hu, “link aggregation refers to a grouping of multiple network links (i.e., communication channels or circuits) into one logical high bandwidth link,” and a “link aggregation group refers to a set or group of physical ports, such as the first plurality of physical ports 122, which are logically grouped to form a single logical port to provide the high-bandwidth link.” *Id.* at 3:19-26. For example, physical ports 122 in FIG. 1 are “associated with a first link aggregation group 124 and can be treated by the content switch 102 as a first logical port.” *Id.* at 3:27-32.

In Hu, the content switch monitors “total bandwidth consumption for each of the multiple ports of the at least one link aggregation group” and rejects a new stream request if the usage would exceed a threshold. Ex. 1009 at 2:35-40, 6:42-47, 6:65-7-8. If the request is accepted, the switch may refer to real-time port usage data and “select a physical port from the link aggregation group based on the real-time port usage data and [] stream a new media stream to a network via the physical port.” *Id.* at 2:50-55, 6:51-64.

Specifically, when a multicast join request is received, a policy module 120 compares the video traffic load at each of the physical ports in the link aggregation group “with a pre-defined threshold.” *Id.* at 5:14-19. “A request module 118 can be executable by the processing logic 104 to determine whether a total video traffic load, including the current video traffic load and a traffic load associated with the requested video stream, would exceed the threshold.” *Id.* at 5:35-39. If the pre-defined threshold would be exceeded, the request for the video stream will be rejected. *Id.* at 5:19-22. If the “total video traffic load, including the current video traffic load and a traffic load associated with the requested video stream” would not exceed the threshold, the content switch may “send the requested video stream via the port having the lowest video traffic load.” *Id.* at 5:31-41. The selection of the physical port may be “based on a stream count associated with the port.” *Id.* at 8:21-24. Thus, the switch makes an “admission control decision” for a multicast request based on current traffic load and anticipated load from the new media stream over ports of the link aggregation group. *Id.* at 7:37-61; *see* Ex. 1003 ¶ [0080].

A stream information module 112 evaluates traffic load on the link aggregation group using various statistics. These statistics may include “a number of join and leave requests associated with the group,” “each join request and each leave request received at the switch/router system since the most recent time measurement point,” “the average bit rate,” “a video stream joined or left, a join or leave time, a bit rate associated with the video stream, an identification of a port affected by the request,” “bandwidth consumption,” and “average traffic load over a period of time, a ratio

of streams to traffic load” Ex. 1009 at 13:8-10, 6:26-32, 7:48-52, 4:3-7, 3:45-49, 5:22-29. Hu further discloses that in dynamically “measuring real-time port usage data,” switch 102 may track the traffic load by increasing and decreasing a count: “For instance, where the measured traffic load at each physical port includes a particular bit rate, the particular bit rate can be increased by a bit rate associated with each join message affecting streams sent via the port since the most recent time measurement point and can be decreased by a bit rate associated with each leave message received affecting streams sent via the port since the most recent time measurement point.” *Id.* at 4:56-63, 6:32-38; *see* Ex. 1003 ¶ [0081].

Hu is analogous art to the '394 Patent. Ex. 1003 ¶ [0082]. First, the '394 Patent and Hu are in the same field of endeavor, namely, developing devices and methods for forwarding data packets in a communications network, particularly multicast packet streams. Ex. 1001 at 1:6-8; Ex. 1009 at Abstract, 3:66-4:3; *see* MPEP § 2141.01(a). Second, Hu is reasonably pertinent to a problem faced by the Upper-Limit Claims of the '394 Patent at least in that both documents seek to control access to multicast packet streams at a network switch. Ex. 1003 ¶ [0082]; Ex. 1001 at 8:36-45, 12:11-32; Ex. 1009 at FIG. 2. For example, the '394 Patent describes setting an upper limit on a number of multicast channels permitted on a switch port and denying a new multicast join request that would exceed the limit (*see* Ex. 1001 at 8:36-9:4), while Hu describes a content switch that will deny a new multicast join request if the request will exceed a pre-defined threshold for a traffic video traffic load on an aggregated port within a Link Aggregation Group (*see* Ex. 1009 at 5:14-41). At least by also seeking to control access to multicast packet streams at a network switch, Hu is analogous art to the '394 Patent. Ex. 1003 ¶ [0082].

E. Ramanathan (Ex. 1010)

U.S. Patent App. Pub. No. 2009/0279549 (“Ramanathan,” Ex. 1010) was filed December 27, 2006, and published November 12, 2009. Accordingly, Ramanathan is prior art to the '394 Patent under 35 U.S.C. §§ 102(a) and 102(e) (pre-AIA). Ramanathan was not cited or otherwise considered during prosecution of the '394 Patent. *See* Ex. 1001 at pp. 1-2; Ex. 1002 at 132, 145-51.

Ramanathan discloses a network switching device 100. As illustrated in its FIG. 1 below, network switching device 100 includes an active processor 102a, a standby processor 102b, and a plurality of line card modules 108a-108d interconnected by a control plane crosspoint 104. Ex. 1010 at FIG. 1.

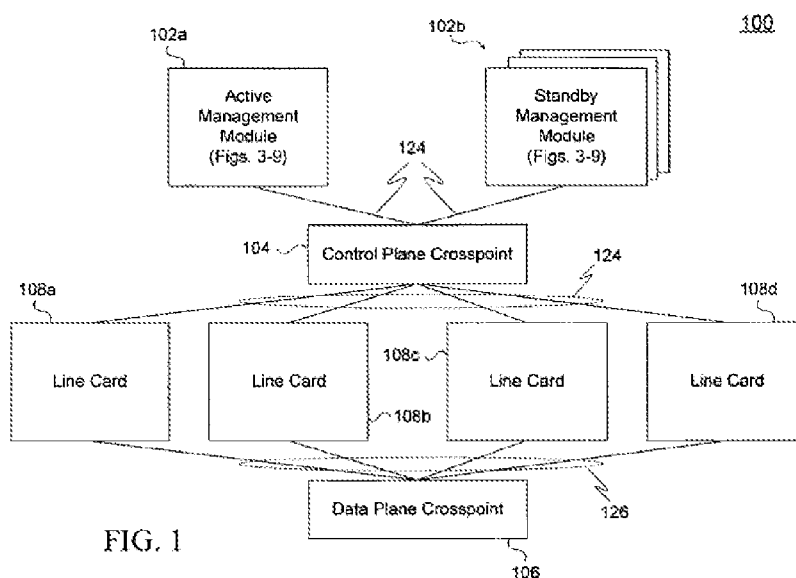


FIG. 1

Each of the line cards includes packet processors 202a-202d and receives or delivers data packets to external devices. *Id.* ¶¶ [0039], [0042]; *see also* Ex. 1003 ¶ [0084]. Another block diagram of network switching device 100, shown below in FIG. 2 of Ramanathan and annotated outside the dashed frame, focuses on only one of the line cards showing each processor “configured with a physical port connector, schematically represented in FIG. 2 by the double arrows, for incoming and outgoing packets.” *Id.* ¶ [0042].

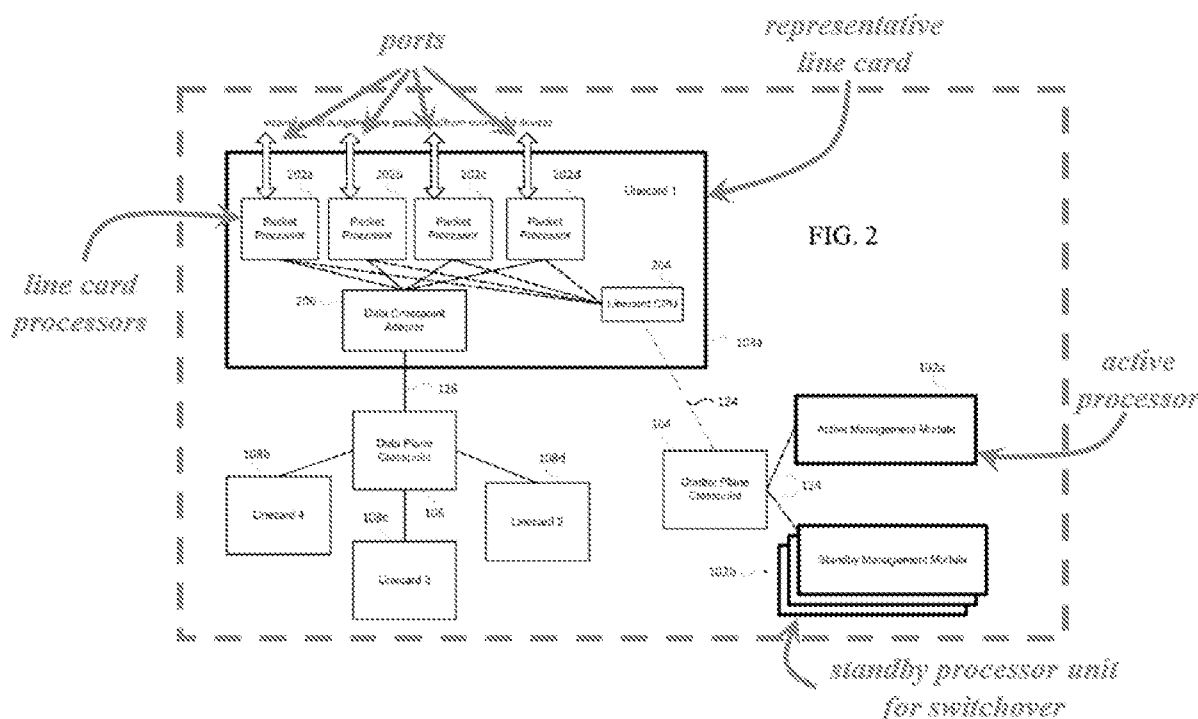


FIG. 2

As illustrated above in annotated FIG. 2, Ramanathan explains that an active processor 102a may be reset through a switchover to standby processor 102b. *Id.* ¶ [0064]. Ramanathan refers to this change as a “hot-switchover.” *Id.* ¶ [0066].

Similarly, one of the line cards, such as line card 1 above, may perform a core reset and be rebooted, or initialized, after its line card processor (LP) 204 receives an upgrade message with an LP image. *Id.* ¶ [0080]; *see* Ex. 1003 ¶ [0085]. Ramanathan refers to the line card in this instance being “hot-inserted.” *Id.* ¶ [0082]. Upon rebooting with the new core, the LP will initialize itself while blocking access to its hardware components. *Id.* ¶ [0081] (“When the LP application is loaded, it will populate its software structures while blocking the access to the HW components during LP initialization, module configuration, and port configuration.”).

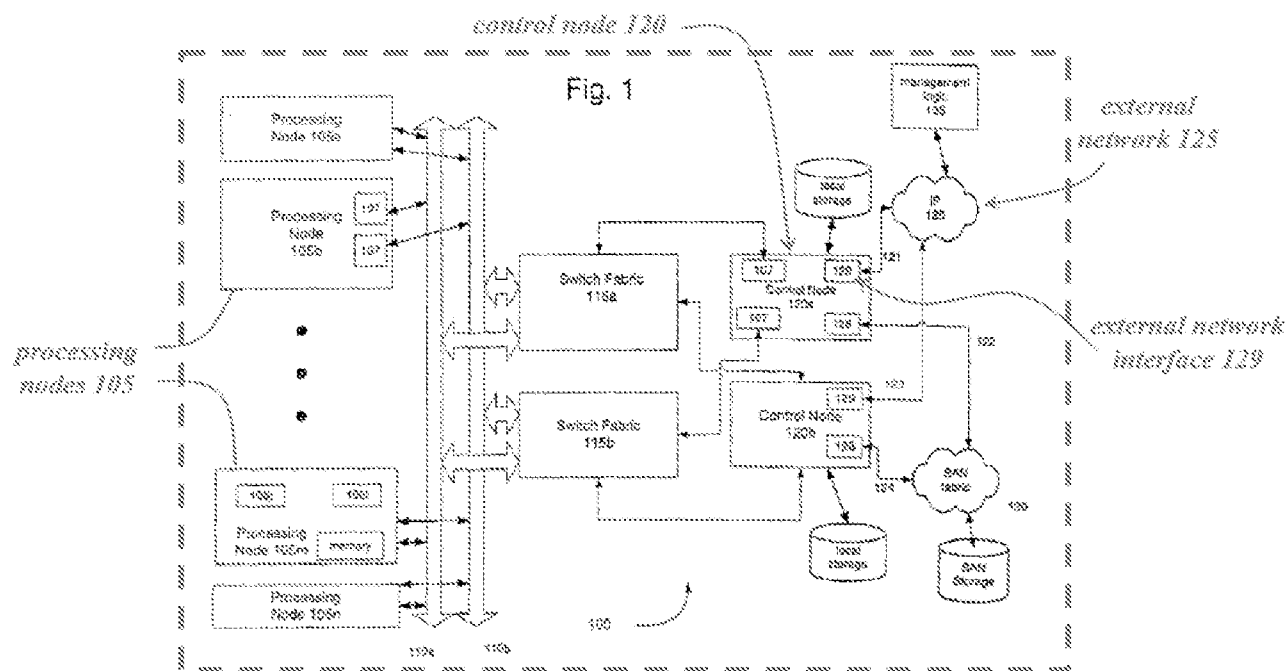
Ramanathan explains the blocking of hardware access during initializing from an LP upgrade or reset in other contexts as well. Ex. 1003 ¶ [0086]. For instance, Ramanathan states that before an LP core reset, MAC table information is stored in a protected portion of memory, and after the LP core reset, the software MAC table is synchronized: “Until this process is completed, all hardware accesses may be disabled.” *Id.* ¶ [0090]. Similarly, With an LP reset, “the VLAN hardware accesses are disabled by the underlying system software. Once the LP application is up, hardware accesses are allowed.” *Id.* ¶ [0092].

Ramanathan is analogous art to the Challenged Claims of the '394 Patent. Ex. 1003 ¶ [0087]. First, the '394 Patent and Ramanathan are in the same field of endeavor, namely, developing devices and methods for forwarding data packets in a communications network, particularly multicast packet streams. Ex. 1001 at 1:6-8; Ex. 1010 ¶¶ [0017], [0039]-[0044], FIG. 2; *see* MPEP § 2141.01(a). Second, Ramanathan is reasonably pertinent to a problem faced by the '394 Patent with respect to the Processor-Initializing Claims at least in that both documents seek to initialize processors within line cards that are “hot-swapped” within an active packet switching network. Ex. 1001 at 6:47-63, 9:41-53, 13:21-33; Ex. 1010 ¶¶ [0080]-[0081]. For example, the '394 Patent describes deactivating ports of a line card when switchover is triggered, such as due to a failure occurring in an active network element (*see* Ex. 1001 at 9:41-53, 11:30-38), while Ramanathan describes performing a core reset to reboot a processor of a line card within an active network by first blocking access to its hardware components (*see* Ex. 1010 ¶¶ [0066], [0080]-[0081]). At least by also seeking to initialize processors within line cards within an active packet switching network, Ramanathan is analogous art to the '394 Patent. Ex. 1003 ¶ [0087].

F. Schulter (Ex. 1011)

U.S. Patent No. 7,174,390 (“Schulter,” Ex. 1011) issued February 6, 2007, from an application filed January 4, 2002, and is prior art to the ’394 Patent under 35 U.S.C. § 102(b) (pre-AIA). Schulter was not cited or otherwise considered during prosecution of the ’394 Patent. *See* Ex. 1001 at pp. 1-2; Ex. 1002 at 132, 145-51.

Schulter is directed to a virtual network system having a hardware platform 100 as generally depicted in FIG. 1 below (annotated outside dashed frame):



Hardware platform 100 includes control nodes 120 functioning as interfaces to the external network 125, as with “line cards 36” in the ’394 Patent. Each control node “is a single board that includes one or more (e.g., 4 processors, local memory, and local disk storage ...) ... that communicates with the Internet (or any other external network) 125 via an external network interface 129 having one or more Gigabit Ethernet NICs connected to Gigabit Ethernet links 121,123.” Ex. 1011 at 3:34-44; *see also* Ex. 1003 ¶ [0089].

Among other things, Schulter teaches detailed steps for swapping out, rebooting, or otherwise initializing processing units. In one example, Schulter discloses “to ‘migrate’ to a new processor” or “to swap a new processor in to a PAN [processing area network] to replace a failing one.” Ex. 1011 at 28:5-12; *id.* at 28:31-32 (“When a new processor is ‘swapped in’ it will need to re-ARP to learn IP address to MAC address associations.”). Another example centers on “failover”

or recovery of failed control nodes, i.e., interface or line card modules, within an active network. *Id.* at 20:1-46.

With respect to failover of a line card, Schulter addresses different operating scenarios for two control nodes such as 120a and 120b. In one scenario, intended to enhance load balancing, both control nodes transmit packets to external nodes, but only one of the control nodes is configured to receive incoming packets. Ex. 1011 at 20:1-3. According to Schulter, [t]he control node doing reception has IP information for filtering and multicast address information for multicast MAC configuration.” *Id.* at 20:7-9. Because this information is needed to process incoming packets, should the receiving control node fail, the transmitting control node must be “failed over” and “configured with all MAC addresses on its physical interface to enable packet reception.” *Id.* at 20:18-22; *see* Ex. 1003 ¶ [0091].

Schulter notes that virtual network drivers keep track of multicast group membership. Therefore, as part of “receive failover,” multicast group membership could be queried from a virtual LAN server “to rebuild the local multicast group membership tables”: “Thus, a receive failover will result in multicast group membership being queried from virtual network drivers to rebuild the local multicast group membership tables.” *Id.* at 20:32-38. According to Schulter, “This operation[] is low overhead and requires no special processing except during failover and recovery, and doesn’t require any special replication of data between control nodes.” *Id.* at 20:38-41; Ex. 1003 ¶ [0092].

Schulter is analogous art to the Challenged Claims of the ’394 Patent. Ex. 1003 ¶ [0093]. First, the ’394 Patent and Schulter are in the same field of endeavor, namely, developing devices and methods for forwarding data packets in a communications network, particularly multicast packet streams. Ex. 1001 at 1:6-8; Ex. 1011 at FIG. 1, 14:25-67; *see* MPEP § 2141.01(a). Second, Schulter is reasonably pertinent to a problem faced by the ’394 Patent with respect to the Processor-Initializing Claims at least in that both documents seek to initialize processors that are “swapped” within line cards within an active packet switching network. Ex. 1001 at 6:47-63, 9:41-53, 13:21-33; Ex. 1011 at 20:1-46, 28:5-12, 28:31-32; *see* Ex. 1003 ¶ [0093]. For example, the ’394 Patent describes initializing a processor of a line card by sending out a status update request and using the responses to update a local filtering database (*see* Ex. 1001 at 9:43-53, 11:30-38), while Schulter describes performing failover of a receiving control node by querying multicast group membership from a networked virtual LAN server to rebuild its local multicast group membership

tables (*see* Ex. 1011 at 20:38-41). At least by also seeking to initialize processors within line cards or control nodes within an active packet switching network, Schulter is analogous art to the '394 Patent. Ex. 1003 ¶ [0093].

III. Substantial New Questions of Patentability

As explained above in Section I.C, the '394 Patent was granted only after the Applicant placed the subject matter of allowable dependent claims in independent form. Namely, the Applicant wrote in independent form the subject matter of application claims 7 and 25 (leading to Upper-Limit Claims 1-7 and 11-17 in the '394 Patent) and the subject matter of application claims 10 and 28 (leading to Processor-Initializing Claims 8-10 and 18 in the '394 Patent). The Examiner identified this incorporated subject matter among the Reasons for Allowance. Ex. 1002 at 238-40.

For the Upper-Limit Claims, Naik (Ex. 1005) in view of new references Reader (Ex. 1008) and Hu (Ex. 1009) presents a new technological teaching of the subject matter of application claims 7 and 25 within independent patent claims 1 and 11, i.e., “wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port, wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.” *See supra* Section I.B.1 (elements [1.7] and [11.7]); Ex. 1003 ¶ [0095]. Because the combination of Naik, Reader, and Hu teaches the subject matter deemed missing from the prior art in the granting of the '394 Patent, “the teaching of the (prior art) patents and printed publications is such that a reasonable examiner would consider the teaching to be important in deciding whether or not the claim is patentable.” *See* MPEP 2242; Ex. 1003 ¶ [0095].

The new technological teaching from the combination of Naik, Reader, and Hu raises a substantial new question of patentability at least for the Upper-Limit Claims 1-7 and 11-17. Had the Examiner been aware of Hu and considered its teachings regarding “aggregating the ports,” “setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port,” “wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port,” and “modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only

when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit,” in combination with the prior art already applied to reject the Upper-Limit Claims before the Applicant’s incorporation of this subject matter, the Examiner would have rejected now-issued claims 1-7 and 11-17 of the ’394 Patent as having been obvious. Ex. 1003 ¶ [0096].

Further, Hu was not cited during prosecution and was not applied in an Office Action. Nor was the art applied (namely, Naik (Ex. 1005), Liu (Ex. 1006), and Deng (Ex. 1007)) cumulative to Hu. While Naik was considered, this Request presents Naik in a new light, namely, in combination with Reader and Hu, which were not previously before the Office, and in light of expert testimony providing background regarding the state of the art that might not have been available to the previous examiner. Ex. 1003 ¶ [0097]. Thus, the “same question of patentability as to the claim has not been decided by the Office in an earlier concluded examination or review of the patent.” MPEP § 2642. In addition, to Requester’s knowledge, the ’394 Patent has not been challenged in a prior post-grant proceeding.

For the Processor-Initializing Claims, Naik in view of new references Reader, Ramanathan (Ex. 1010), and Schulter (Ex. 1011) presents a new technological teaching of the subject matter of application claims 10 and 28 within independent patent claims 8 and 18, i.e., “wherein operating the network element comprises initializing a new processing unit added to the network element by: deactivating the ports to which the new processing unit is assigned; while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request, receiving multicast membership reports sent by multicast clients in response to the multicast status request, constructing the list by the new processing unit based on the received multicast membership reports; and after constructing the list, activating the ports to which the new processing unit is assigned.” *See supra* Section I.B.2 (elements [8.6] and [18.6]); Ex. 1003 ¶ [0098]. Because the combination of Naik, Reader, Ramanathan, and Schulter teach the subject matter deemed missing from the prior art in the granting of the ’394 Patent, “the teaching of the (prior art) patents and printed publications is such that a reasonable examiner would consider the teaching to be important in deciding whether or not the claim is patentable.” *See* MPEP 2242; Ex. 1003 ¶ [0098].

The new technological teaching from the combination of Naik, Reader, Ramanathan, and Schulter raises a substantial new question of patentability at least for the Processor-Initializing Claims 8-10 and 18. Had the Examiner had been aware of Ramanathan and Schulter and

considered their teachings regarding “initializing a new processing unit added to the network element” in combination with the prior art already applied to reject the claims before the Applicant’s incorporation of this subject matter, the Examiner would have rejected now-issued claims 8-10 and 18 of the ’394 Patent as having been obvious. Ex. 1003 ¶ [0099].

Further, Ramanathan and Schulter were not cited during prosecution and were not applied in an Office Action. Nor was the art applied (namely, Naik (Ex. 1005), Liu (Ex. 1006), and Deng (Ex. 1007)) cumulative to Ramanathan and Schulter. Thus, the “same question of patentability as to the claim has not been decided by the Office in an earlier concluded examination or review of the patent.” MPEP § 2642. In addition, to Requester’s knowledge, the ’394 Patent has not been challenged in a prior post-grant proceeding.

In light of these substantial new questions of patentability, Requester requests reexamination of the Challenged Claims based on the following Grounds:

Challenge	Claims	Statutory Ground	Prior Art References
Ground 1	1-7, 11-15, and 17	35 U.S.C. § 103 (pre-AIA)	• Naik (Ex. 1005) • Reader (Ex. 1008) • Hu (Ex. 1009)
Ground 2	16	35 U.S.C. § 103 (pre-AIA)	• Naik (Ex. 1005) • Reader (Ex. 1008) • Hu (Ex. 1009) • Deng (Ex. 1007)
Ground 3	8, 9, and 18	35 U.S.C. § 103 (pre-AIA)	• Naik (Ex. 1005) • Reader (Ex. 1008) • Ramanathan (Ex. 1010) • Schulter (Ex. 1011)
Ground 4	10	35 U.S.C. § 103 (pre-AIA)	• Naik (Ex. 1005) • Reader (Ex. 1008) • Ramanathan (Ex. 1010) • Schulter (Ex. 1011) • Deng (Ex. 1007)

IV. Detailed Application of the Prior Art to Every Claim for Which Reexamination Is Requested

In Grounds 1-4 below, Requester explains in detail and on an element-by-element basis how the prior art references listed above establish the unpatentability of the Challenged Claims under 35 U.S.C. § 103(a) (pre-AIA). This Request is supported by a declaration of a technical expert, Anthony Wechselberger (Ex. 1003).

For readability, Requester has highlighted claim language in bold. Requester has also added in italics any other emphasis to the text in this Request unless otherwise noted.

Requester respectfully requests the issuance of an Office Action rejecting the Challenged Claims based on one or more of Grounds 1-4.

A. Upper-Limit Claims

1. Ground 1: Naik in View of Reader Further in View of Hu Would Have Rendered Obvious Claims 1-7, 11-15, and 17

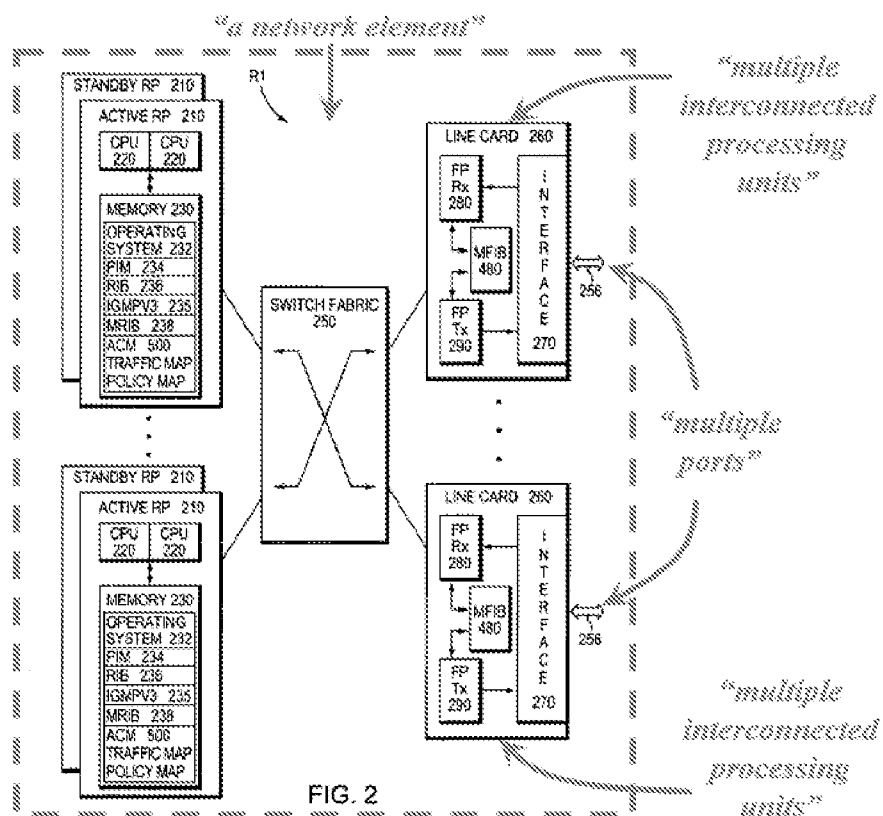
a. Independent Claim 1

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

For the preamble, Naik discloses a **method for communication** (FIG. 1), as claimed. Ex. 1005 at Abstract; *see also id.* ¶ [0001] (“This invention relates generally to computer networks, and more particularly, to controlling multicast classes of traffic destined for hosts or end users in a computer network.”); Ex. 1002 at 189; Ex. 1003 ¶ [0103].

ii. [1.1] “operating a network element, [1.1.1] wherein the network element further comprises multiple interconnected processing units and multiple ports,”

Naik discloses or suggests elements [1.1] and [1.1.1]. Ex. 1003 ¶ [0104]. In particular, Naik discloses **operating a network element** (router operating system 232 on edge router R1), **wherein the network element further comprises multiple interconnected** (switch fabric 250) **processing units** (line cards 260) **and multiple ports** (ports P1-P4 in FIG. 1 and ports 256 in FIG. 2). *Id.*



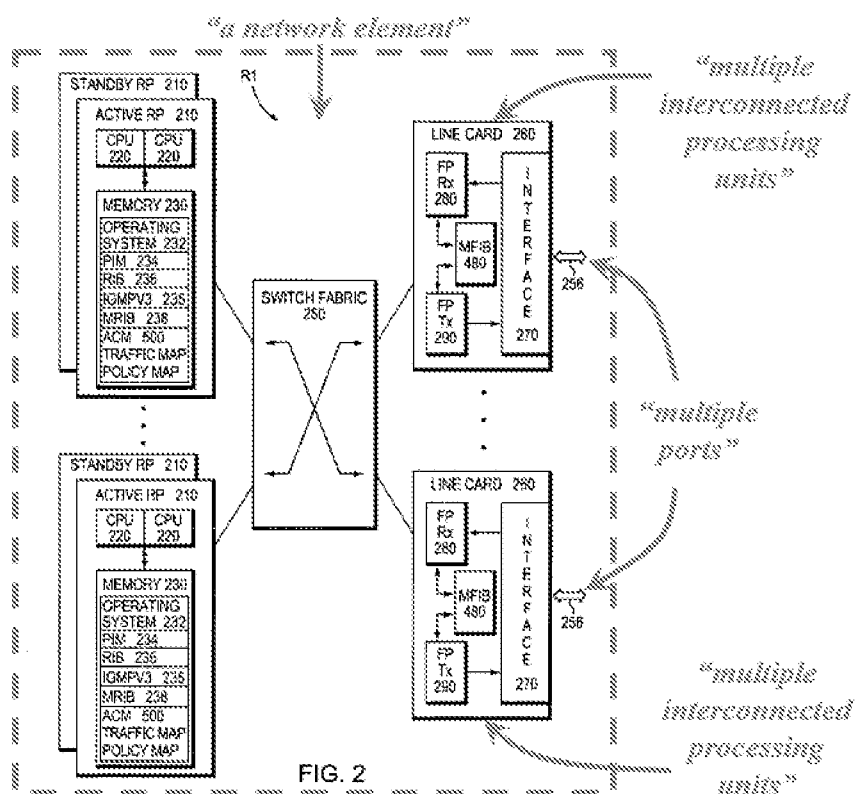
As shown above in FIG. 2 (annotated outside the dashed frame), edge router R1 corresponds to the claimed network element. Ex. 1003 ¶ [0105]. The edge router is configured as a multicast router and, as illustrated in FIG. 1 of Naik, may be situated as an intermediate node within a computer network to route data traffic between end nodes, such as hosts or users. Ex. 1005 ¶¶ [0002]-[0003]. Edge router R1 may also be connected to a subnetwork of a local area network (LAN). *Id.* ¶ [0004].

As also annotated above, edge router R1 includes multiple line cards (or interface cards) 260 interconnected through switch fabric 250, together corresponding to the claimed “**multiple interconnected processing units.**” Ex. 1003 ¶ [0106]. Edge router R1 also includes route processors 210 interconnected through switch fabric 250. According to Naik, “line cards 260 perform forwarding of the data traffic, while the route processors 210 handle routing and control of the data traffic forwarding.” Ex. 1005 ¶ [0046]. Switch fabric 250, or “interconnect fabric 250” may be, for example, “a crossbar interconnection switch or high-speed bus,” and corresponds to the claimed “**interconnected**” feature of the multiple processing units in edge router R1. *Id.* ¶ [0040].

Shown above in FIG. 2, edge router R1 includes ports 256 (also shown as P1-P4 in FIG. 1) providing input/output paths for the device and corresponding to the claimed “**multiple ports.**” *Id.* ¶ [0046]. Ports 256 are coupled to “transmitting and receiving interface 270 electronics” within each of line cards 260. *Id.*

iii. [1.1.2] “wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and”

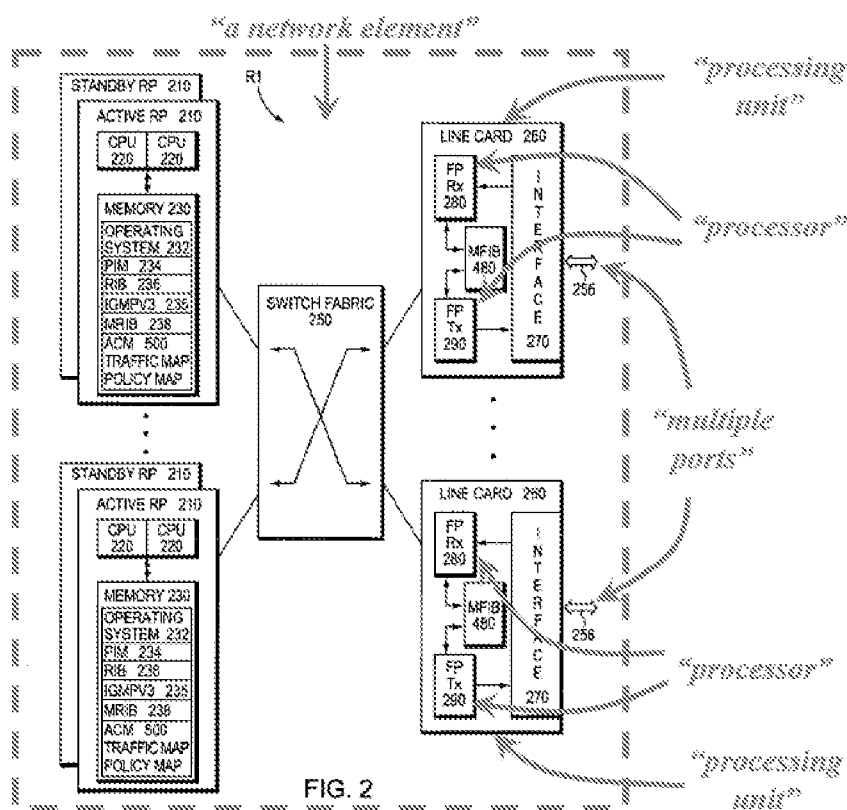
Naik discloses or suggests element [1.1.2]. Ex. 1003 ¶ [0108]. Namely, Naik discloses wherein each processing unit (line cards 260) is assigned to a different subset of the multiple ports (subset of ports 256 in FIG. 1 associated with each line card 260) of the network element (edge router R1). *Id.*



In edge router R1, ports 256 correspond to the claimed “**multiple ports**” in element [1.1.2] and are divided into subsets. *Id.* ¶ [0109]. As shown in FIG. 2 above (annotated outside the dashed frame), “each processing unit” as claimed (i.e., each line card 260) is “assigned to a different subset of the multiple ports” labeled 256 for edge router R1. Ex. 1005 ¶ [0046]; Ex. 1002 at 189. This is because each line card 260 has its own set of “a plurality of ports 256 with transmitting and receiving interface 270 electronics.” Ex. 1005 ¶ [0046].

- iv. [1.1.3] “wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned;”

Naik discloses or suggests the subject matter of element [1.1.3], i.e., **wherein each processing unit** (line cards 260) **contains a processor** (receive forwarding processor (FP Rx) and transmit forwarding processor (FP Tx) within each line card 260) **for processing packets** (data traffic) **communicated over the different subset of the multiple ports** (subset of ports 256 in FIG. 1 associated with each line card 260) **to which the processing unit** (one of line cards 260) **is assigned**. Ex. 1003 ¶ [0110].



As shown above in FIG. 2 (annotated outside dashed frame), each line card 260, corresponding to the claimed “**each processing unit**” contains a processor in the form of a receive forwarding processor (FP Rx) and a transmit forwarding processor (FP Tx). These processors make forwarding decisions for incoming and outgoing packets for edge router R1. *Id.* ¶ [0046] According to Naik, “[t]he FP Rx renders a forwarding decision for each packet received at a port and determines where to forward the packet.” *Id.* “Likewise, the FP Tx performs lookup operations (using MFIB) on a packet transmitted from the router via one or more ports.” Thus, in Naik, **each**

processing unit (line card 260) **contains a processor** (processors FP Rx and/or FP Tx), as claimed. Ex. 1003 ¶ [0111].

Further, each of Naik's processors is **for processing packets communicated over the different subset of the multiple ports** (ports 256) **to which the processing unit** (line card 260) **is assigned**. *Id.* ¶ [0112]. This is because, on each line card 260, "[t]he interface [270] has a plurality of ports 256," and "the interface 270 is coupled to a receive forwarding processor (FP Rx) and a transmit forwarding processor (FP Tx)" to receive and forward "data traffic." Ex. 1005 ¶ [0044].

Accordingly, both receive forwarding processor (FP Rx) and transmit forwarding processor (FP Tx) in each line card 260 corresponds to the "**processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned**," recited in element [1.1.3]. Ex. 1003 ¶ [0113].

- v. [1.2] "**storing in each processing unit a respective list, [1.2.1] each list indicating all of one or more multicast packet streams that are permitted for forwarding by the network element, and**"

Naik discloses or suggests the subject matter of elements [1.2] and [1.2.1]. Ex. 1003 ¶ [0114]. Particularly, Naik discloses **storing in each processing unit** (line cards 260) **a respective list** (Multicast Forwarding Information Base (MFIB) 480 and its MFIB Forwarding Table 400), **each list indicating all of one or more multicast packet streams that are permitted for forwarding** (entries 410 within MFIB Forwarding Table 400 in FIG. 4) **by the network element** (edge router R1)." *Id.*

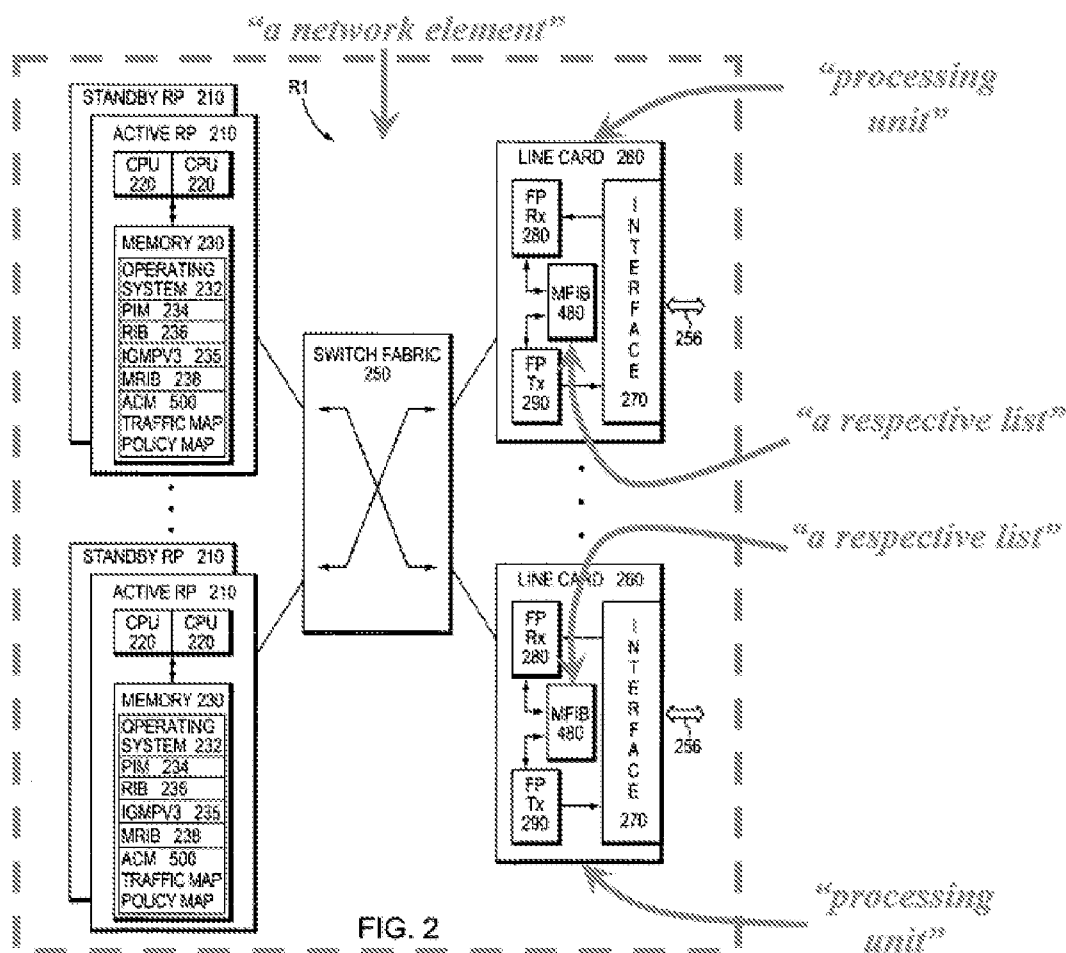


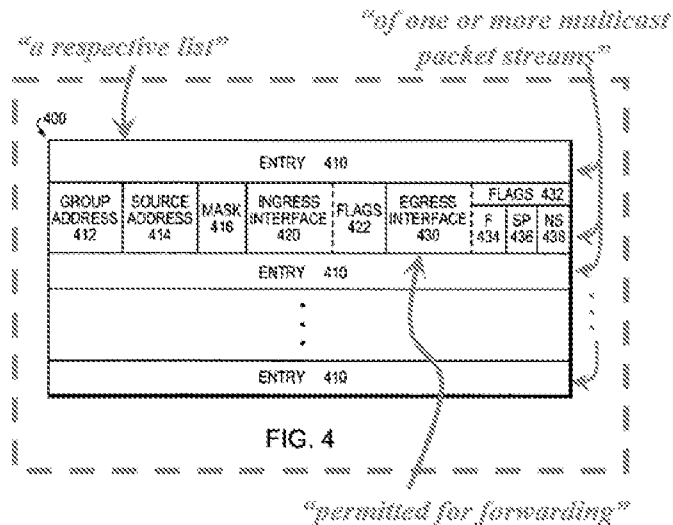
FIG. 2

First, Naik's line cards 260 necessarily have memory for "**storing a respective list**" (MFIB 480 and its MFIB Forwarding Table 400), as claimed. Ex. 1003 ¶ [0115]. Annotated FIG. 2 above (annotations outside dashed frame) shows that each line card 260 includes a Multicast Forwarding Information Base (MFIB) 480, labeled identically as "MFIB 480," which corresponds to the claimed "**respective list**" for "**each processing unit**," i.e., each line card 260. Each MFIB 480 via its forwarding table 400, shown in FIG. 4, indicates all of one or more multicast packet streams that are permitted for forwarding by the network element (edge router R1). Ex. 1005 ¶ [0045]; Ex. 1003 ¶ [0115].

As explained in Naik, route processors 210 essentially build MFIB 480. Specially, a Multicast Route Information Base (MRIB) 238 is stored in memory 230 and executed by a route processor 210 (*see* FIG. 2 above, at left). Ex. 1005 ¶ [0045]. Edge router R1 exchanges packet routing information with its neighbors in the network and "builds the MRIB 238." *Id.* The MRIB then provides its routing information, from which the MFIBs, i.e., the "**respective list**" in element

[1.2], is derived. *Id.* Using information from the MRIB, the MFIBs are “populated” on each of line cards 260: “The MFIB 480 is derived from the MRIB and is embodied *as one or more multicast forwarding tables 400 that are populated on the line cards 260 and whose contents describe how to forward data packets through the router.*”). *Id.* Moreover, as shown in FIG. 2, Naik indicates that each line card 260 contains an identical forwarding table 400 by indicating “MFIB 480” as the same element in every line card 260, not for example as “MFIB 480-1 ... MFIB 480-N.” Ex. 1003 ¶ [0116].

Second, for element [1.2.1], as shown in FIG. 4 below (annotated outside of dashed frame), an MFIB Forwarding Table 400 contains various information “**indicating all of one or more multicast packet streams that are permitted for forwarding by the network element,**” as claimed (edge router R1):



Specifically, each entry 410 in the table of FIG. 4 represents a data packet. *Id.* ¶ [0050]. Moreover, the fields within each entry 410 indicate information relevant to the forwarding of a packet by the network element (edge router R1). *Id.* For instance, the illustrated fields within MFIB Forwarding Table 400 include a multicast group address 412, an ingress interface 42, and an egress interface 430, indicating permission for forwarding the packet. *Id.* ¶¶ [0050]-[0051]. The group address field 412 contains “a multicast group network (IP) address,” and egress interface 430 field contains “a list of output (forwarding) interfaces over which the incoming packet should be forwarded.” *Id.* ¶ [0051].

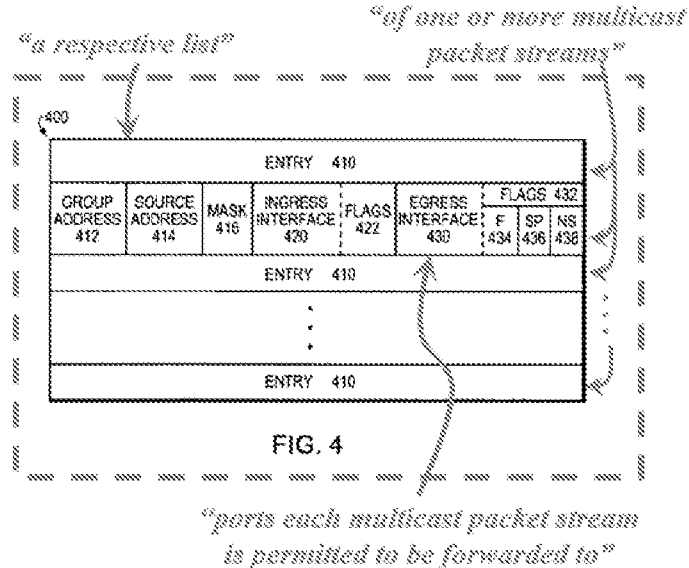
Therefore, multiple rows in FIG. 4 correspond to the claimed “**one or more multicast packet streams.**” And the presence of an entry 410 within MFIB Forwarding Table 400, together

with the data within group address field 412 and egress interface field 430, correspond to a list indicating “**all of one or more packet streams that are permitted for forwarding by the network element,**” as recited in element [1.2.1]. *Id.*; *see also id.* ¶ [0052] (“In response to receiving the incoming packet from a neighbor, the MFIB 480 performs a lookup into its forwarding table 400 to find a route of an entry 410 that matches a multicast destination address of the packet. The matching route instructs the router as to which interfaces the package should be forwarded.”); Ex. 1003 ¶ [0118]; Ex. 1002 at 189.

Therefore, Naik discloses or suggests every limitation of elements [1.2] and [1.2.1]. Ex. 1003 ¶ [0119].

vi. [1.2.2] “**further indicating which of the one or more ports each multicast packet stream is permitted to be forwarded to;**”

Naik discloses or suggests the content of element [1.2.2]. Ex. 1003 ¶ [0120]. That is, Naik’s MFIB 480 with MFIB Forwarding Table 400 **further indicat[es] which of the one or more ports** (one of the one or more ports 256 in FIG. 2) **each multicast packet stream** (collection of entries 410 in MFIB Forwarding Table 400) **is permitted to be forwarded to** (egress interface 430). *Id.*



As shown in Naik FIG. 4 above (annotated outside dashed frame), MFIB Forwarding List 400 includes an egress interface field 430. Egress interface field 430 contains “a list of output (forwarding) interfaces over which the incoming packet should be forwarded.” Ex. 1005 ¶ [0046]. The egress/output interfaces are ports within the edge router through which a packet stream may be forwarded to another device in the network, such as another node. *Id.* ¶ [0010] (“On each node,

packets are received on the ingress interface towards the root of the tree and packets *are forwarded out egress interfaces* towards the receivers or nodes.”). A POSITA would understand that “egress interfaces” in a network switch are synonymous with “ports,” such as ports 256 from which multicast packets exit edge router R1. Ex. 1003 ¶ [0121]; *id.* ¶ [0046] (“[T]he FP Tx performs lookup operations (using MFIB) *on a packet transmitted from the router via one or more ports.*”). Naik further explains that the entries within MFIB Forwarding Table 400 indicate which ports each multicast packet stream “**is permitted to be forwarded to,**” as claimed, when the MFIB 480 performs a lookup into table 400: “In response to receiving the incoming packet from a neighbor, the MFIB 480 performs a lookup into its forwarding table 400 to find a route of an entry 410 that matches a destination address of the packet. The matching route instructs the router as to which interfaces the packet should be forwarded.” Ex. 1005 ¶ [0052]; Ex. 1003 ¶ [0121].

As a result, at least egress interface field 430 corresponds to “**further indicating which of the one or more ports each multicast packet stream is permitted to be forwarded to,**” as recited in element [1.2.2], and Naik discloses or suggests all limitations of this element. Ex. 1003 ¶ [0122]; Ex. 1002 at 189.

vii. [1.3] “receiving from a client a request relating to reception of a given multicast packet stream;”

Naik further teaches **receiving from a client** (end user, Step 708 in FIG. 7) **a request relating to reception of a given multicast packet stream** (a multicast request within IGMPv3 packets 504, such as an IGMP Join request), as recited in element [1.3]. Ex. 1003 ¶ [0123].

Naik discloses an admission control module (ACM) 500 that snoops, or monitors, “incoming request[s] issued by the end nodes to join, leave or modify membership in multicast groups designated to receive the data.” Ex. 1005 ¶¶ [0023], [0053]. From these end nodes, “end users” or “**clients**” may submit the multicast requests to edge router R1 in the form of IGMP join and leave packets within a Membership Report. *Id.* ¶¶ [0024], [0063], FIGS. 6B-6C.

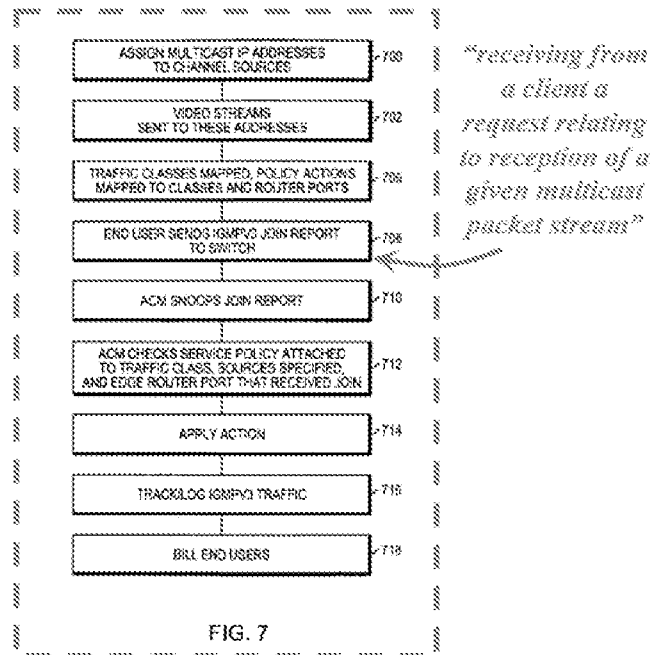


FIG. 7, annotated above outside the dashed frame, is a flow diagram of a method disclosed in Naik about whether to accept or deny a streaming request. Ex. 1005 ¶¶ [0067], [0073]. Step 708 indicates “End User Sends IGMPv3 Join Report to Switch.” *Id.* ¶ [0073]. Because the multicast request is a client request to join a multicast stream, receiving that request equates to “**receiving from a client a request relating to reception of a given multicast packet stream,**” as recited in element [1.3]. Ex. 1003 ¶ [0125]; Ex. 1002 at 190.

viii. [1.4] [1.4.1] “**distributing the request among the multiple processing units within the network element, and**”

Naik discloses element [1.4.1] in several ways. Ex. 1003 ¶ [0126].

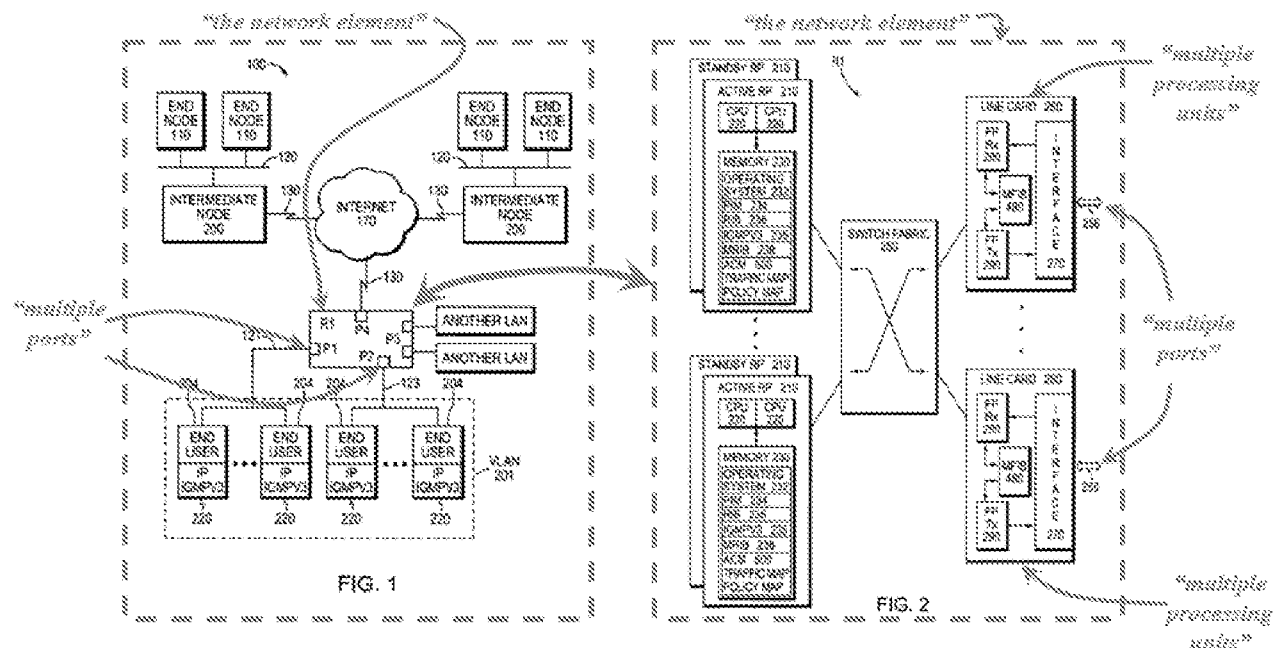
First, Naik discloses **distributing the request** (edge router R1 forwards a received multicast request to its neighboring routers on the link in the direction of the rendezvous point or multicast source) **among the multiple processing units** (line cards 260 including interface 270 and ports 256 function as outputs to the neighboring routers) **within the network element** (edge router R1). *Id.*

The phrase “**distributing the request**” of element [1.4.1] at least includes a router forwarding a multicast request to the source of the multicast packet stream within the network. *Id.* ¶ [0127]. The ’394 Patent confirms this claim scope through element [3.1] of claim 3, which depends from claim 1 and specifies: “wherein distributing the request comprises ... forwarding [a]

first copy [of the request] to a source of the given multicast packet stream, ...” See *Alcon Research, Ltd. v. Apotex Inc.*, 687 F.3d 1362, 1367 (Fed. Cir. 2012) (“It is axiomatic that a dependent claim cannot be broader than the claim from which it depends.... Therefore if claim 2 covers the range from 0.0001% w/v-5% w/v, claim 1 must cover at least that range.”).

Naik discloses that forwarding step, explaining that a router receiving a multicast request “communicates that request [i.e., “**distributing the request**”] to its neighboring routers (neighbors) on the link towards the rendezvous point (for a shared tree) or source for a SPT [shortest path tree],” from which “the routers construct a multicast distribution tree rooted at a rendezvous point or source for that group address.” Ex. 1005 ¶ [0012]. Thus, in at least this way, Naik discloses “**distributing the request**” in element [1.4.1]. Ex. 1003 ¶ [0128].

Further, forwarding of the received multicast request to “neighboring routers” towards the rendezvous point or link, as taught by Naik, necessarily requires distributing the request “**among the multiple processing units with the network element**,” as claimed. As illustrated in Naik, edge router R1 generally includes ports P1-P4 for interfacing with other nodes in computer network 100 (FIG. 1, left, with annotations outside the dashed frame), which are implemented as ports 256 in each of the line cards 260 (FIG. 2, right, with annotations outside the dashed frame):



Id. ¶¶ [0036], [0066]. Ports P1-P4 are implemented in router R1 in line cards 260 in FIG. 2 as “a plurality of ports 256 with transmitting and receiving interface 270 electronics.” *Id.* ¶ [0046]. These electronics include a receive forwarding processor (FP Rx) that “performs lookup operations

(using MFIB) on a packet transmitted from the router via one or more ports.” *Id.* Consequently, to be forwarded from edge router R1 to “neighboring routers (neighbors) on the link,” as taught by Naik, the forwarded multicast request must be distributed “**among the multiple processing units**,” i.e., among the one or more line cards 260, in order to exit router R1. *Id.* ¶ [0128].

Second, in another context, Naik discloses the subject matter of element [1.4.1]. Ex. 1003 ¶ [0130]. In particular, Naik discloses **distributing the request** ((i) multicast request arrives, (ii) MFIB executing on line cards 260 notifies route processors 210 of arrival, (iii) Admission Control Module (ACM) 500 executing on route processors 210 snoops information from the multicast request, and (iv) Multicast Route Information Base (MRIB) executing on route processors 210 populates the MFIB 480 and MFIB Forwarding Tables 400 based on the multicast request) **among the multiple processing units** (line cards 260 executing MFIB 480 and MFIB Forwarding Tables 400) **within the network element** (edge router R1). *Id.*

As Naik explains, in a first step, a receive forwarding processor (FP Rx) within line card 260 “renders a forwarding decision for each packet received at a port and determines where to forward the packet.” *Id.* ¶ [0046]. “To that end, the FP RX makes use of a multicast forwarding information base (MFIB) 480” (*id.*), and the MFIB notifies “the route processor whenever an IGMPv3 packet arrives...” (*id.* ¶ [0052]). Further, according to Naik, line card 260 includes an internal forwarding database to help distribute the packet to a route processor 210: “In the event that the packet is to be forwarded to one of the router’s two route processors, the FP RX makes use of an internal forwarding information base (IFIB) to determine to which route processor the packet should be forwarded.” *Id.* ¶ [0046].

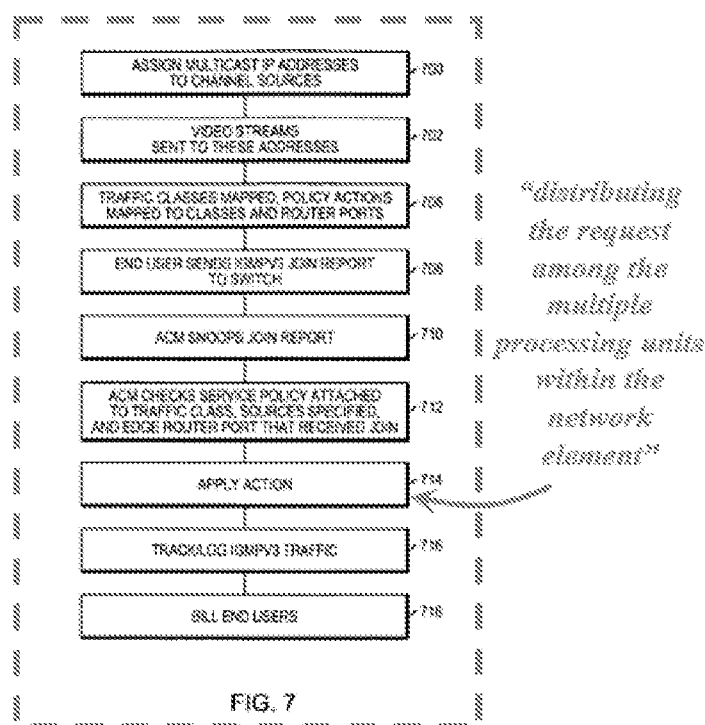
After the receiving line card 260 has forwarded the multicast request to one or more of the route processors 210, the Admission Control Module (ACM) 500 executing on those route processors snoops packets of the request to extract its information in a second step. Ex. 1005 ¶ [0052] ([T]he ACM is configured to snoop all such arriving packets.”). That snooping involves extracting certain information and classifying the extracted information according to traffic map 506 within memory 230. *Id.* at FIG. 2 ¶¶ [0045], [0055], [0070]. Based on a configurable policy map 508, the ACM decides “whether the IGMPv3 request should be accepted or denied.” *Id.* ¶ [0067]; *see also id.* ¶¶ [0023], [0025], [0053], [0055].

If the ACM accepts the multicast request, the routing and forwarding tables must be updated so that the multicast packet stream can be delivered properly. Ex. 1003 ¶ [0133].

Accordingly, the router updates a Multicast Route Information Base (MRIB) 238 with routing information of multicast streams. Ex. 1005 ¶ [0045]. Then, the MRIB, operating in the route processors 210, “**distribut[es] the request**,” as claimed, by “populat[ing]” the MFIB 480 and MFIB Forwarding Tables 400 of the line cards with forwarding information corresponding to the multicast request. Ex. 1005 at FIG. 4, ¶ [0045] (the MFIB derived from the MRIB “is embodied as one or more multicast forwarding tables 400 that are populated on the line cards 260 and whose contents describe how to forward data packets through the router.”); Ex. 1003 ¶ [0133].

This “**distributing the request**” occurs “**among the multiple processing units**,” as recited in element [1.4.1], because MFIB 480 and MFIB Forwarding Tables 400 are in each of the line cards 260 (i.e., “**processing units**”) within edge router R1. Ex. 1005 at FIG. 2; Ex. 1003 ¶ [0134]. And these “**multiple processing units**,” i.e., line cards 260, are “**within the network element**,” i.e., within edge router R1, as claimed. Ex. 1005 at FIG. 2, ¶ [0045]. *Id.* ¶ [0134].

Naik further discloses element [1.4.1], “**distributing the request among the multiple processing units within the network element**,” in the flow chart of FIG. 7 below (annotated outside the dashed frame):



As shown above, after an end user sends a multicast request (Step 708), the ACM snoops the request (Step 710), and the ACM checks traffic and policy maps (Step 712), route processors

210 executing ACM 500 will “Apply Action” to the requesting port. Ex. 1005 ¶ [0073] (“The ACM snoops the IGMPv3 packets received and if a join message [multicast request] is detected at 710, the ACM ... checks the policy map at 712. If an action is found, it is automatically *applied to the requesting port* 714.”). Indeed, even if the decision of the ACM is to deny admission to the multicast group, updating of MFIB Forwarding Table 400 in line cards 260 still occurs: “If the action is to deny admission, the requestor does not become a member of the group, or if already a member the requestor *is removed* from the group and/or *from the forwarding table*.” *Id.* ¶ [0074]. This updating of MFIB Forwarding Table 400 in line cards 260 is necessary for the line cards 260 to properly forward the multicast packet stream that is the subject of the “**request**.” Ex. 1003 ¶ [0136].

Therefore, in several ways, Naik discloses or suggests “**distributing the request among the multiple processing units within the network element**,” as recited in element [1.4.1]. *Id.* ¶ [0137].

ix. [1.4.2] “updating the respective list in each processing unit responsively to the request; and”

Naik, alone or in view of Reader, discloses or suggests all limitations of element [1.4.2]. Ex. 1003 ¶ [0138].

Naik. As found by the Examiner (*see* Ex. 1002 at 190-91), Naik discloses or suggests **updating the respective list** (Multicast Forwarding Information Base (MFIB) 480 and its MFIB Forwarding Table 400) **in each processing unit** (line cards 260) **responsively to the request** (in response to receiving a multicast request within IGMPv3 packets 504, such as an IGMP Join request), as recited in element [1.4.2]. Ex. 1003 ¶ [0139].

In Naik, the multicast forwarding tables 400 (the “**respective list**”) are populated (“**updating**”) on the line cards 260 (“**each processing unit**”) after ACM 500 accepts a multicast request and routing information is updated in MRIB 238 (“**responsively to the request**”):

[T]he router operating system 232 implements multicast components as separate software process modules that are distributed over active route processors 210 and line cards 260 which, in turn, are individually hot-swappable and may have redundant (standby) backups. To that end, the multicast modules include Multicast Forwarding Information Bases (MFIB’s) 1-N 480 executing on the line cards and multicast components including the RIB 236, PIM 234, IGMPv3 235[,] ACM 500, and a MRIB 238 executing on the route processors 210. Operationally, *the router*

builds the MRIB 238 . . . after exchanging routing information with the neighbors. . . . The primary role of the MRIB 238 is to facilitate communication between the various multicast components, i.e., coordinate distribution of state between the protocol applications (e.g., PIM, IGMPv3, etc.) and between these protocols and MFIB. As a result, MRIB provides the routing information and a multicast-capable path to each multicast destination. The MFIB 480 is derived from the MRIB and is embodied as one or more multicast forwarding tables 400 that are populated on the line cards 260 and whose contents describe how to forward data packets through the router.

Ex. 1005 ¶ [0045]; *see also id.* ¶ [0054] (“The ACM 500 includes a process 502 that snoops 502 the IGMPv3 packets 504, e.g. the joins and reports from all end users, and classifies the traffic via a traffic map 506” and “monitor[s] and track[s] the specific end users and the ports through which the joins traveled.”). Moreover, a POSITA would have understood that forwarding tables 400 must be updated to reflect the acceptance of a new multicast group member so that the packet stream will be properly forwarded by the router. Ex. 1003 ¶ [0140]. Therefore, Naik discloses or suggests the subject matter of element [1.4.2]. *Id.*

Reader. To the extent Naik is viewed as not explicitly disclosing **updating the respective list in “each” processing unit responsively to the request**, Reader discloses this subject matter. Ex. 1003 ¶ [0141]. In particular, Reader teaches **updating the respective list** (slave multicast forwarding databases (SM-FDBs)) **in each processing unit** (all existing interfaces 210A-210N, 230, 240) **responsively to the request** (in response to updates for MM-FDB 350 due to authorized IGMP membership report messages). *Id.*

Reader discloses a LAN switch (“**the network element**”) that includes network interfaces 210A-210N, 230 and 240 implemented by “a combination of discrete logic and programmable logic, such as programmable network processors,” which correspond to the claimed “**each processing unit.**” Ex. 1008 ¶ [0018], FIG. 2; Ex. 1003 ¶ [0142]. For forwarding IP Multicast data packets according to network policy, a switch manager 250 within LAN switch 150 maintains a master multicast forwarding database (MM-FDB) 350 and a multicast authorization database (M-ADB) 340. Ex. 1008 ¶¶ [0019], [0021], [0024], [0025], FIG. 3. The master multicast forwarding database (MM-FDB) 350 operating in switch manager 250 “has as entries multicast groups and associated switch ports through which destination hosts that are registered in the multicast groups are reachable.” *Id.* ¶ [0021].

When one of interfaces 210A-210N, 230, and 240 snoops incoming packets and identifies an IGMP membership report, i.e., a multicast request (**“the request”**), the switch checks with the authorization database M-ADB 340 to see if the requesting port “is authorized to join the multicast group” by network policy. *Id.* ¶ [0025]. If the request is authorized, switch manager 250 “updates [**“updating”**] MM-FDB 350 [a master copy of **“the respective list”**] to include the new multicast group/port association [**“the request”**].” *Id.* Then, switch manager 250 distributes the contents of MM-FDB 350 to *all* the interfaces in the device to **update** the slave multicast forwarding databases (SM-FDBs) (**“the respective list in each processing unit”**):

Switch manager 250 distributes the contents of MM-FDB 350 [including **“the request”**] to interfaces 210A through 210N, 230, 240 [**“each processing unit”**] in response to updates for MM-FDB 350 [**“responsively to the request”**] and thereby maintains [**“updating”**] slave multicast forwarding databases (SM-FDBs) [**“the respective list”**] on interfaces 210A through 210N, 230, 240.

Id. ¶ [0021].

Therefore, Reader discloses or suggests **“updating the respective list in each processing unit responsively to the request,”** as recited in element [1.4.2]. Ex. 1003 ¶ [0144].

Motivation to Combine. A POSITA would have been motivated to implement in edge router R1 of Naik the process of updating a forwarding list in each line card or interface responsively to a multicast request, as disclosed in Reader, with a reasonable expectation of success. Ex. 1003 ¶ [0145]. This motivation and expectation of success would have existed at least because updating the forwarding table or list in each line card would have provided uniform forwarding data at all ports, resulting in more accurate and faster routing of multicast packet streams compared with when forwarding tables at the line cards differ, which accuracy and speed a POSITA would have wanted to achieve. *Id.*

The references provide teaching, suggestion, and motivation for combining their teachings in this manner. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398 (2007); MPEP § 2143.I. For example, to the extent Naik does not expressly disclose MFIB 480 and MFIB Forwarding Table 400 being the updated the same way in each of line cards 260, Naik at least suggests that arrangement. In FIG. 2, Naik illustrates the Multicast Forwarding Information Base 480 with the identical box and label “MFIB 480” in every line card 260, not for example as “MFIB 480-1 ... MFIB 480-N,” which might denote a difference between those databases. Further, Naik teaches that MRIB 238 “is embodied as one or more multicast forwarding tables 400 that are *populated on the line cards*

260,” implying that a change to the lists is—or at least may be—provided to “**each processing unit**,” as claimed. These disclosures provide at least a suggestion to treat forwarding tables 400 alike, which Reader further discloses by updating a common forwarding database (SM-FDB) with forwarding data reflecting a new multicast group request on *all* interfaces of the device. Ex. 1003 ¶ [0146]; Ex. 1005 ¶ [0045]; Ex. 1008 ¶ [0021].

A POSITA would also have been motivated to have combined the teachings of Naik and Reader because the combination would have involved the use of a known technique to improve a similar device in the same way. Ex. 1003 ¶ [0147]; MPEP § 2143 at I(C). Under this analysis, Naik discloses a “base” device in the form of edge router R1 having ACM 500 that applies a policy map 508 to restrict admission for certain multicast requests. Ex. 1005 ¶ [0055]. An MRIB 238 containing multicast routing data, executing on the route processors, “populate[s]” forwarding tables 400 within line cards 260. *Id.* ¶ [0045]. Reader presents a comparable device in the form of LAN switch 150 employing authentication server 140 that checks whether to allow access to multicast traffic requested by an end user according to a network policy. Ex. 1008 ¶ [0004]. Reader “has been improved in the same way as the claimed invention” (MPEP § 2143 at I(C)), because it “distributes the contents” of a master multicast forwarding database (MM-FDB) 350 to slave multicast forwarding databases (SM-FDBs) on all interfaces of the device “in response to updates to MM-FDB 350 and thereby maintains slave multicast forwarding databases (SM-FDBs)” on all the interfaces. Ex. 1008 ¶ [0021]; Ex. 1003 ¶ [0147].

Further, implementing the “improvement” of Reader in the edge router of Naik would have been technically minor for a POSITA, involving minor variations in software for router processors 210 to push updates to forwarding tables 400 in all line cards 260 whenever a change to MRIB 238 occurs from an accepted multicast request. Ex. 1003 ¶ [0148]. Following this implementation, the modification of edge router R1 in Naik to be **updating the respective list in each processing unit responsively to the request**, would have yielded a predictable and desirable result of having forwarding tables 400 on all line cards 260 contain the same information, so that those line cards could accurately and rapidly forward multicast traffic regardless of which line card receives or sends the packets. *Id.*

For at least these reasons, Naik and Reader disclose or suggest all limitations of element [1.4.2], and Naik in view of Reader would have rendered obvious “**updating the respective list**

in each processing unit responsively to the request,” as recited in element [1.4.2]. Ex. 1003 ¶ [0149].

- x. **[1.5] “selectively forwarding multicast packets associated with the given multicast packet stream by the processing units in accordance with the respective updated lists,”**

Naik, or Naik in view of Reader, discloses or suggests the subject matter of element [1.5]. Ex. 1003 ¶ [150].

Naik. Naik discloses **selectively forwarding multicast packets** (“pushing the packets” of a multicast channel to an end user over the selected route) **associated with the given multicast packet stream** (a requested/joined multicast channel) **by the processing units** (line cards 260) **in accordance with the respective updated lists** (MFIB 480 performs a lookup into its forwarding table 400 to find a route of an entry 410), as claimed. *Id.*

As explained in Naik, after ACM 500 snoops a received multicast request, “[t]he actions include accepting membership in a group associated with a multicast channel [i.e., “**the given multicast packet stream**”] and pushing the packets to the end user [i.e., “**selectively forwarding multicast packets associated with the given multicast packet stream**”].” Ex. 1005 at Abstract. Naik further explains the router’s function of selectively forwarding packets based on updated lists, such as forwarding tables 400: “In response to receiving the incoming packet from a neighbor, the MFIB 480 performs a lookup into its forwarding table 400 to find a route of an entry 410 that matches a multicast destination address of the packet [i.e., “**in accordance with the respective updated lists**”]. The matching route instructs the router as to which interfaces the packet should be forwarded [i.e., “**selectively forwarding**”].” *Id.* ¶ [0052] Naik also explains the forwarding of packets in accordance with forwarding tables updated in response to a multicast request in the context of internet TV: “The user via the set-top-box sends a IGMP join to the edge router which creates the necessary PIM routing states [i.e., “**in accordance with the respective updated lists**”] and starts pushing the video streams to the end user.” *Id.* ¶ [0020]; *see also id.* ¶¶ [0073]–[0084], FIG. 7 (discussing tracking and logging of multicast packets in video streams after snooping and accepting a multicast request).

Further, according to Naik, electronics within line cards 260 (**the processing units**) perform the selective forwarding: “[T]he line cards 260 [i.e., “**the processing units**”] perform forwarding of the data traffic [i.e., “**selectively forwarding**”], ... The interface 270 is coupled to

a receive forwarding processor (FPRX) and a transmit forwarding processor (FPTx)... [T]he FPTx performs lookup operations (using MFIB) [i.e., “**in accordance with the respective updated lists**”] on a packet transmitted from the router [i.e., “**multicast packets associated with the given multicast packet stream**”] via one or more ports.” Ex. 1005 ¶ [0046]; Ex. 1003 ¶ [0152].

In addition, a POSITA would have understood that selectively forwarding multicast packets associated with the given multicast packet stream by the processing units in accordance with the respective updated lists, according to [1.5], was required to abide by the new multicast membership following the multicast request. Ex. 1003 ¶ [0153]. Without selective forwarding by the processing units (or line cards) based on the respective updated forwarding tables in those line cards, the multicast channel may not be delivered to the user as requested. *Id.*

Reader. Moreover, to the extent Naik is viewed as not explicitly disclosing the subject matter of element [1.5], Reader discloses that content. Ex. 1003 ¶ [0154]. For instance, Reader discloses **selectively forwarding multicast packets** (data packet transmitted on all resolved switch ports) **associated with the given multicast packet stream** (group of IP Multicast data packets) **by the processing units** (the ingress one of interfaces 210-210N, 230, 240 for the data packet) **in accordance with the respective updated lists** (resolve group address in slave multicast forwarding database SM-FDB). *Id.*

More particularly, Reader teaches that switch manager 250 within LAN switch 150 maintains a master multicast forwarding database (MM-FDB) 350, while the multiple interfaces (“**the processing units**”) contain slave multicast forwarding databases (SM-FDBs) (“**respective updated lists**”). Ex. 1008 ¶ [0021]; Ex. 1003 ¶ [0155]. The master MM-FDB 350 contains a list of “multicast groups and associated switch ports through which destination hosts that are registered in the multicast groups are reachable.” Ex. 1008 ¶ [0021]. When changes occur to the master MM-FDB 350, such as the acceptance of an authorized multicast request within an IGMP membership report, the MM-FDB 350 is updated “to include the new multicast group/port association.” *Id.* ¶ [0025]. In response to this update, switch manager 250 “distributes the contents of MM-FDB 350 to interfaces 210A through 210N, 230, 240 ... and thereby maintains slave multicast forwarding databases (SM-FDBs) [i.e., “**respective updated lists**”] on interfaces 210A through 210N, 230, 240 [i.e., “**the processing units**”]. *Id.* ¶ [0021].

Thereafter, LAN switch 150 **selectively forward[s]** multicast packets according to these **respective updated lists**:

In IP Multicast forwarding on LAN switch 150, the SM-FDB [**“respective updated lists”**] on the ingress one of interfaces 210A through 210N, 230, 240 [**“the processing units”**] is invoked to resolve a multicast group address [**“the given multicast packet stream ... in accordance with the respective updated lists”**] in an IP Multicast data packet [**“multicast packets”**] to one or more switch ports, and the data packet is transmitted on all resolved switch ports [**“selectively forwarding”**], except the ingress switch port if it is one of the resolved switch ports.

Id.

Thus, Naik, or Naik in view of Reader, discloses or suggests the subject matter of element [1.5]. Ex. 1003 ¶ [0157].

Motivation to Combine. Further to the discussion about combining the teachings of Naik and Reader in element [1.4.2] above, a POSITA would have been motivated to implement in edge router R1 of Naik the process of selectively forwarding multicast packets for a multicast packet stream by the processing units in accordance with the respective updated lists as disclosed in Reader with a reasonable expectation of success at least because doing so would have ensured accurate distribution of packets within a multicast channel to the end user that submitted a multicast request. Ex. 1003 ¶ [0158].

Naik discloses that receive forwarding processor (FP Rx) in each line card 260 “renders a forwarding decision for each packet received at a port and determines where to forward the packet” using MFIB 480 and that transmit forwarding processor (FP Tx) “performs lookup operations (using MFIB) on a packet transmitted from the router via one or more ports.” Ex. 1005 ¶ [0046]. Because these forwarding processors in each line card 260 rely on MFIB 480, a POSITA would have been motivated to selectively forward multicast packets by the processing units in accordance with respective updated lists so their decisions on “where to forward the packet” will be accurately executed, particularly according to the new multicast membership following the multicast request. Ex. 1003 ¶ [0159].

Relying on **“respective updated lists”** for selectively forwarding packets would also have yielded predictable results and a reasonable expectation of success for a POSITA at least because an admission control module in both references—whether ACM 500 applying policy map 508 in Naik or multicast authorization database (M-ADB) 340 applying network policy in Reader—filters out inappropriate multicast requests and ensures that updated forwarding databases include only acceptable or authorized new members. *Id.*; Ex. 1003 ¶ [0160]; *see also supra* Section IV.A.1.a.ix.

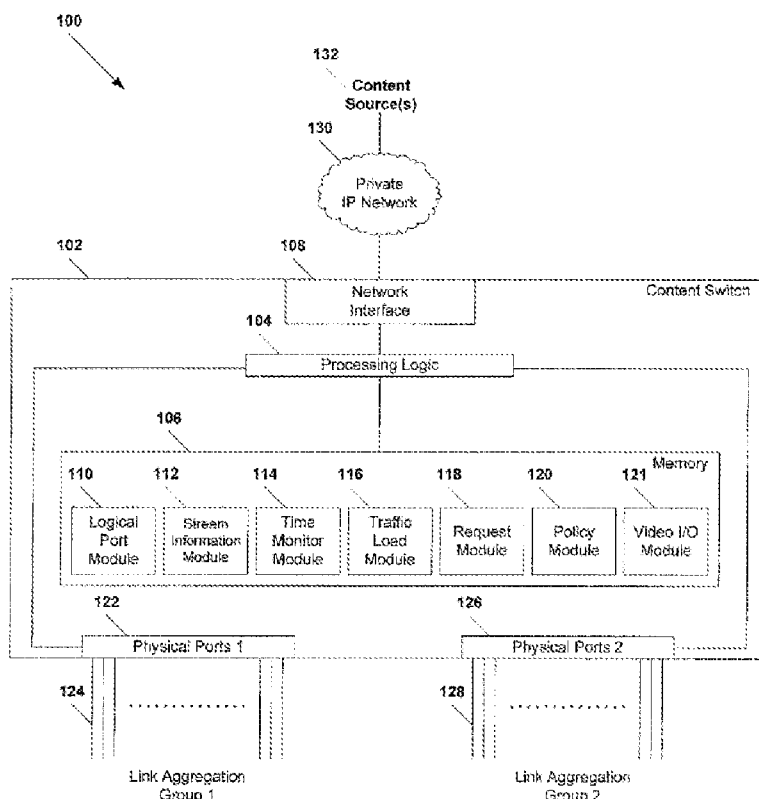
Therefore, Naik, or Naik in view of Reader, discloses or suggests all limitations of element [1.5], and would have rendered obvious “**selectively forwarding multicast packets associated with the given multicast packet stream by the processing units in accordance with the respective updated lists,**” as recited in element [1.5]. Ex. 1003 ¶ [0161].

- xi. [1.6] “**wherein operating the network element comprises aggregating two or more of the ports to form a high-capacity port, and**”

Naik and Hu disclose or suggest all limitations of element [1.6]. Ex. 1003 ¶ [0162].

Naik and Hu. With respect to element [1.6], Naik discloses operating a **network element** (edge router R1) comprising **two or more ports** (ports P1-P4 in FIG. 1 or ports 256 in FIG. 2) on a **processing unit** (line cards 260). Naik does not expressly disclose or suggest “**wherein operating the network element comprises aggregating two or more of the ports to form a high-capacity port,**” as recited in element [1.6]. But Hu discloses or suggests **wherein operating the network element** (content switch 102) **comprises aggregating** (aggregating into Link Aggregation Group 1 (124) or Link Aggregation Group 2 (128) in compliance with IEEE 802.3ad standard) **two or more of the ports** (physical ports 122 or physical ports 126 in FIG. 1, respectively) **to form a high-capacity port** (a single logical port to provide the high-bandwidth link). *Id.*

Hu is directed to a method of allocating media streams on a network content switch of an Internet Protocol Television (IPTV) network or other packet-based video distribution network. Ex. 1009 at 2:56-61. A content switch 102 (“**network element**”), as shown in FIG. 1 below, within system 100 has “processing logic 104, such as one or more independent or redundant processors,” memory 106, and physical ports such as 122 and 126 each logically grouped into a “link aggregation group” (“**aggregating two or more of the ports to form a high-capacity port**”). *Id.* at 3:7-19; Ex. 1003 ¶ [0163].

**FIG. 1**

According to Hu, “link aggregation refers to a grouping of multiple network links (i.e., communication channels or circuits) into one logical high bandwidth link [i.e., “**high-capacity port**”], which may be used to route audio data, text data, video data, or any combination thereof.” *Id.* at 3:19-22. Hu discloses that a “link aggregation group refers to a set or group of physical ports [i.e., “**two or more of the ports**”], such as the first plurality of physical ports 122, which are logically grouped to form a single logical port [i.e., “**aggregating**”] to provide the high-bandwidth link [i.e., “**to form a high-capacity port**”]” (*id.* at 3:23-26), and that physical ports 122 in FIG. 1 are “associated with a first link aggregation group 124 and can be treated by the content switch 102 as a first logical port” (*id.* at 3:27-32).

Thus, Naik in view of Hu discloses or suggests the subject matter of element [1.6]. Ex. 1003 ¶ [0163].

Motivation to Combine. A POSITA would have been motivated to implement Hu’s teachings to aggregate two or more ports to form a high-capacity port into Naik’s edge router R1 with a reasonable expectation of success because aggregating the ports to form a link aggregation

port would have provided higher bandwidth while using lower speed ports of edge router R1 and improved load balancing. Ex. 1003 ¶ [0164].

Naik and Hu provide a teaching, suggestion, and motivation for their combination. *See* MPEP § 2143.I; Ex. 1003 ¶ [0165]. For example, Naik addresses the expansion of multicast streaming into television-type video: “Multicast has found its way into cable TV, where providers are sending ‘video over IP’ to end users. This is typically achieved using the Ethernet-to-the-home (ETTH) deployment where video, voice and data are carried over a single Ethernet media to the end users.” Ex. 1005 ¶ [0019]. The reference notes some of the challenges in that implementation, such as flooding individual ports. *Id.* ¶ [0020] (“The user via the set-top-box sends a IGMP join to the edge router which creates the necessary PIM routing states and starts pushing the video streams to the end user. Access lists ... are defined to accept/deny and IGMP request, but such an implementation will send multicast traffic to all edge router ports that are connected to a given VLAN, possibly flooding the ports.”). As a result, a POSITA would have been motivated to find improvements to multicasting of video streams, and Hu provides one in a similar system, as “system 100 includes a content switch 102, such as a network content switch/router system or other edge system of an Internet Protocol Television (IPTV) network or other packet-based video distribution network.” Ex. 1009 at 2:57-61; Ex. 1003 ¶ [0165].

Hu also provides express motivation to incorporate its teachings related to link aggregation. Ex. 1003 ¶ [0166]. Hu explains that a link aggregation group provides “much higher bandwidth via lower speed ports” than when the ports are separated. Ex. 1009 at 1:31-34. Because Naik aims to route multicast traffic, especially video traffic, through a plurality of ports, obtaining “much higher bandwidth via lower speed ports” would be a natural attraction to a POSITA seeking to improve on the teachings in Naik and to expand the usefulness of edge router R1. Ex. 1003 ¶ [0166]. This higher bandwidth could be achieved using the existing “lower speed ports,” thereby avoiding physical changes to edge router R1 and capitalizing on existing structure. *Id.*

Similarly, Naik addresses the efficient routing of packets and focuses on setting limits to avoid excessive loads on its ports (Ex. 1005 ¶¶ [0007], [0072]), while Hu indicates that link aggregation can help “increase overall utilization efficiency of network components and quality of service to end-users” by increasing load balancing of packet traffic (Ex. 1009 at 1:13-15). *See also id.* at 1:31-34 (“Such link aggregation ... enables distribution of network traffic across the aggregated ports to achieve load balancing.”) Accordingly, a POSITA would have wanted to

investigate improving load balancing for Naik's edge router to increase "utilization efficiency" and "quality of service" according to the suggestions in Hu, leading to consideration of link aggregation among physical ports 256 in edge router R1 of Naik. Ex. 1003 ¶ [0167].

A POSITA also would have had a reasonable expectation of success from implementing the teachings of Hu into edge router R1 of Naik because "**aggregating two or more of the ports to form a high-capacity port**" is standardized. Ex. 1003 ¶ [0168]. Hu identifies the IEEE 803.ad standard as the basis for link aggregation:

[L]ink aggregation allows multiple physical links of a switch/router to be combined into a logical port that provides much higher bandwidth via lower speed ports and enables distribution of network traffic across the aggregated ports to achieve load balancing. A technique for communicating link aggregation information between switches/ routers is described in an industry standard (i.e., the IEEE 803.ad Standard), which describes a protocol for communicating link aggregation information.

Ex. 1009 at 1:30-39. By implementing an industry standard, a POSITA would have reasonably expected that implementing link aggregation as taught by Hu into the router of Naik would have yielded predictable results. Further, following IEEE 802.3ad would have had the benefit for a POSITA of ensuring compatibility and interoperability of the Naik-Hu combination with the IEEE standard. Ex. 1003 ¶ [0168].

Similarly, following the guidance in the MPEP, combining the teachings of Naik and Hu would have involved "combining prior art elements according to known methods to yield predictable results." MPEP § 2143.I(A). First, Naik and Reader disclose each element of claim 1 discussed above, and Hu discloses the subject matter of this element [1.6], with the only difference "being the lack of actual combination of the elements in a single prior art reference." *Id.*; Ex. 1003 ¶ [0169]. Second, a POSITA could have combined these prior-art elements by known methods. Ex. 1003 ¶ [0169]. For example, a POSITA, possibly consulting IEEE 802.3ad, would have "**aggregate[ed] two or more of the ports to form a high-capacity port**" by joining multiple physical ports within ports 256 of Naik into a common logical link, as taught by Hu, together with making minor changes to control software in the processors of Naik. *Id.*

These modifications would have involved well-known techniques to implement a standardized feature on packet switches such as edge router R1. *Id.* First, a POSITA would have modified edge router R1 in Naik for "aggregating the ports" and forwarding streams "over the

high-capacity port” taught by Hu by following IEEE 802.3ad and forming a logical link between multiple physical ports. Second, the POSITA would have updated software within the ports and line cards 260 of Naik to process the changed mapping relationships resulting from link aggregation. *Id.* Thus, consulting IEEE 802.3ad and joining multiple physical ports within ports 256 of Naik into a common logical link, as taught by Hu, together with making minor changes to control software in the processors of Naik, would have enabled a POSITA to implement aggregated ports as recited in element [1.6] in an edge router disclosed in Naik. Ex. 1003 ¶ [0170].

And when combined, each element would “perform[] the same function as it does separately.” MPEP § 2143.I(A). The “processing units” would continue to consult forwarding tables to identify the path for packets and ports 256 would continue to provide an interface to the “processing units” for ingress and egress of packets, although software updates would be required to process the one or more logical ports formed from existing physical ports within the “high-capacity port.” Ex. 1003 ¶ [0171]. Finally, as discussed above, a POSITA would have recognized the results of the proposed combination as being predictable, which is evidenced at least by the processing of “aggregating two or more of the ports to form a high-capacity port” being standardized by IEEE and being taught by Hu to be applied to devices for switching packets within a communication network, as with edge router R1 in Naik. *Id.*

Accordingly, Naik and Hu disclose or suggest all limitations of element [1.6], and the combination of Naik, Reader, and Hu would have rendered obvious “**wherein operating the network element comprises aggregating two or more of the ports to form a high-capacity port,**” as claimed. Ex. 1003 ¶ [0172].

xii. [1.7] “**wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port,**”

Naik and Hu disclose or suggest all limitations of element [1.7]. Ex. 1003 ¶ [0173].

Naik. Naik discloses or suggests element [1.7] except for “aggregating the ports” and “over the high-capacity port.” *Id.* That is, as explained below, Naik discloses **setting an upper limit** (“LIMIT” action to prevent DOS attacks) **on a number of the multicast packet streams** (maximum number of “joins” allowed by the switch) **that are permitted for forwarding** (DENY 510 “denying admittance to a group once a limit number of requests to join ... is reached”). *Id.*

Naik uses a traffic map 506 and a policy map 508 for **setting an upper limit on a number of packet streams permitted for forwarding over a port**, as claimed. In particular, Naik discloses that, in snooping multicast requests, the actions of admission control module (ACM) 500 consults a traffic map 506 and a policy map 508. *See* Ex. 1005 ¶ [0054]. Traffic map 506 includes a setting for “the maximum number of hosts allowed on a port, and the maximum number of groups or channels allowed on a specific port,” which corresponds to “**setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over**” a port, as recited in element [1.7]. *Id.* ¶¶ [0025], [0055]; *id.* at Abstract (“Incoming packet traffic received by an edge router is snooped and, when a request to join a multicast channel is received, the traffic is analyzed” and “the actions may be to limit the number of routing states, by denying admittance to a groups [sic] once a limit number of requests to join, or other such parameter, is reached.”); Ex. 1003 ¶ [0174].

Naik explains that setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over a port can help in deterring denial of service (DOS) attacks. *Id.* ¶¶ [0023], [0053] (“The ACM also prevents attacks, such as denial of service (DOS) on the intermediate node by limiting the number of joins and leave states for the multicast group.”). As described in Naik, “DOS attacks try to create large streams of joins/leaves that will cause the PIM to create large routing structures or tie up the processors. If unchecked such attacks will eventually crash the system.” *Id.* ¶ [0072]. Therefore, the ACM in Naik stores an upper limit on a number of multicast packet streams for a port:

The ACM, in an illustrative embodiment, would be loaded with a predetermining a [sic, predetermined] maximum number of “joins” allowed by the switch, or “joins” received from a port, or the number of ports subscribed for a channel, or combinations thereof or of other relevant parameters.

Id.; Ex. 1003 ¶ [0175].

Hu. Naik, however, does not expressly disclose setting the upper limit on a number of multicast packet streams as part of “**aggregating the ports**” or that the upper limit is for streams “**that are permitted for forwarding over the high-capacity port.**” Hu teaches that subject matter. Ex. 1003 ¶ [0176].

Hu discloses **wherein aggregating the ports** (physical ports 122 associated with Link Aggregation Group 1 (124) and physical ports 126 associated with Link Aggregation Group 2

(128) in FIG. 1) **comprises setting an upper limit** (pre-defined threshold) **on a number of the multicast packet streams** (video traffic loads including “a number of join and leave requests associated with the group”) **that are permitted for forwarding over the high-capacity port** (Steps 202-208 in FIG. 2). Ex. 1003 ¶ [0177].

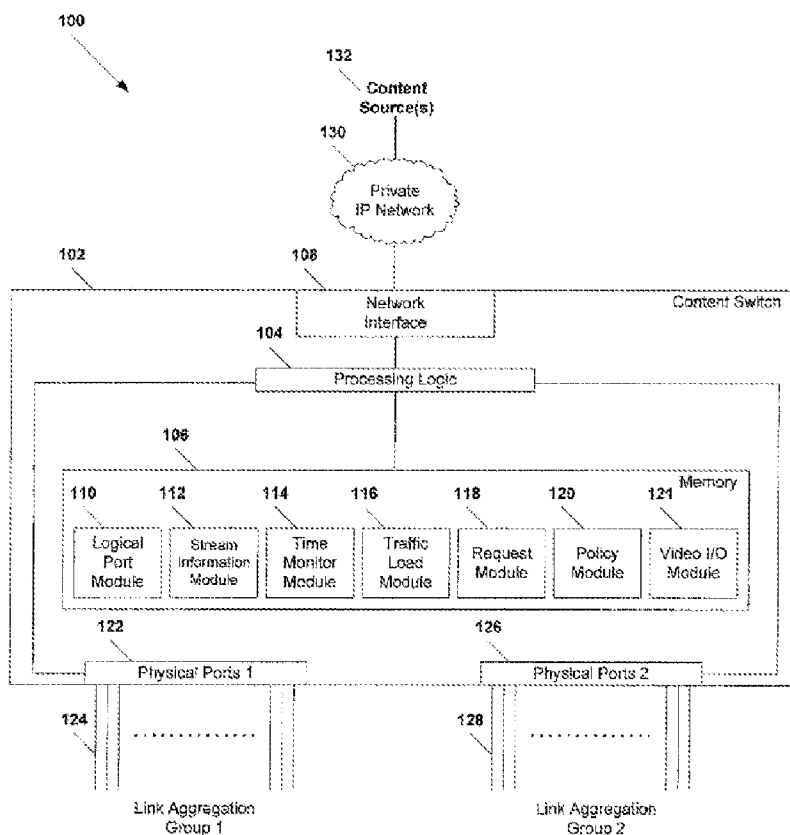


FIG. 1

Hu is directed to a method of allocating media streams on a network content switch of an Internet Protocol Television (IPTV) network or other packet-based video distribution network. Ex. 1009 at 2:56-61. A content switch/router 102, as shown in FIG. 1 above, within system 100 has “processing logic 104, such as one or more independent or redundant processors,” memory 106, and physical ports such as 122 and 126. *Id.* at 3:7-19.

First, Hu groups physical ports 122 into Link Aggregation Group 1 (124) and physical ports 126 into a Link Aggregation Group 2 (128), each corresponding to “**aggregating the ports**” into “**the high-capacity port**,” as recited in element [1.7]. *Id.* at 3:12-32. According to Hu, “link

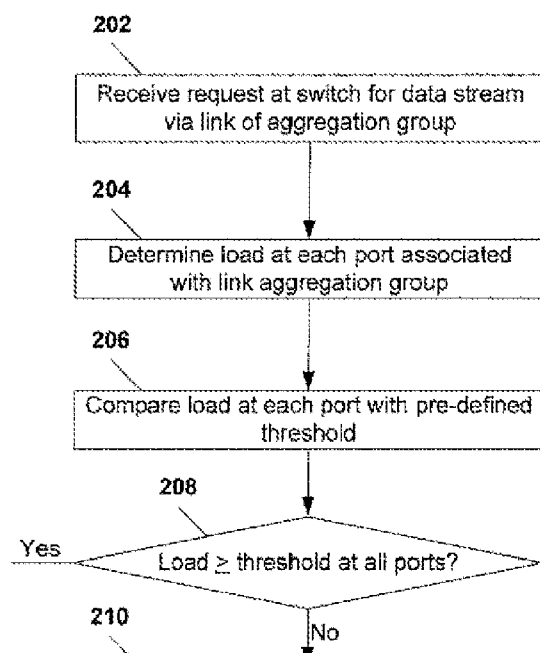
aggregation refers to a grouping of multiple network links (i.e., communication channels or circuits) into one logical high bandwidth link, which may be used to route audio data, text data, video data, or any combination thereof.” *Id.* at 3:19-22. Hu discloses that a “link aggregation group refers to a set or group of physical ports, such as the first plurality of physical ports 122, which are logically grouped to form a single logical port to provide the high-bandwidth link” (*id.* at 3:23-26), and that physical ports 122 in FIG. 1 are “associated with a first link aggregation group 124 and can be treated by the content switch 102 as a first logical port” (*id.* at 3:27-32).

Second, Hu discloses “**setting an upper limit on the number of the multicast packet streams that are permitted for forwarding over the high-capacity port**” by setting a “pre-defined threshold” on video traffic loads. Ex. 1003 ¶ [0180]. Specifically, a policy module 120, shown in FIG. 1 above, stores “a pre-defined threshold” that can be used to block traffic from requests for new streams:

“In a particular embodiment, the memory 106 can include a policy module 120 that *stores data indicating threshold video traffic loads*. The policy module 120 can be executable by the processing logic 104 to compare the video data traffic load at each of the first plurality of physical ports 122 with a pre-defined threshold. *If video traffic loads at all of the first plurality of physical ports 122 exceed the pre-defined threshold, policy module 120 can be executable by the processing logic 104 to reject the request for the video stream.*”

Ex. 1009 at 5:14-22.

Moreover, Figure 2 of Hu, reproduced in part below, shows Hu’s threshold (**upper limit**) is set as part of port aggregation. Ex. 1003 ¶ [0181]. That is, in Hu, “[**the**] **aggregating the ports comprises**” setting the threshold, as recited in element [1.7]:



As shown above, switch/router 102 receives requests via a link of an aggregation group to join a data stream and evaluates whether the traffic load at the relevant ports would exceed the pre-defined threshold. The specification in Hu makes clear that setting the threshold occurs within a broader method of aggregating ports:

Moving to block 204, *the switch/router system determines a traffic load at each physical port associated with the link aggregation group through which the request was received. ...* Proceeding to block 206, *the switch/router system compares the traffic load at each port with a pre-defined threshold stored at the switch/router system.* Continuing to decision node 208, *the switch/router system determines whether the traffic loads at all of the ports associated with the link aggregation group meet or exceed the pre-defined threshold.*

Id. at 6:23-44; Ex. 1003 ¶ [0182]. Hu further explains that “traffic load” for which the pre-defined threshold is set can include, for example, “a number of join and leave requests associated with the group,” “a ratio of streams to traffic load,” or “each join request and each leave request received at the switch/router system since the most recent time measurement point.” *Id.* at 13:8-10, 5:22-29.

Consequently, Hu discloses or suggests “**wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port,**” as recited in element [1.7]. Ex. 1003 ¶ [0183].

Motivation to Combine. Further to the discussion about combining the teachings of Naik and Hu in element [1.6] above, a POSITA would have been motivated to extend the teachings of Naik with respect to setting an upper limit on the number of multicast streams permitted for forwarding across a port to a high-capacity port as part of a link aggregation group as taught by Hu with a reasonable expectation of success at least because Hu teaches that doing so can lead to allocation of media streams between ports in the standardized link aggregation group for achieving load balancing. Ex. 1003 ¶ [0184]; Ex. 1009 at 1:6-42, 2:13-30.

Relative to element [1.7], Naik and Hu contain several additional teachings, suggestions, and motivations for the combination. *See KSR*, 550 U.S. 398; MPEP § 2143.I. Naik, for example, teaches to monitor and limit traffic load on a port because a number of channels or a number of users beyond an upper limit may evidence a Denial of Service (DOS) attack on the device. According to Naik, these attacks can debilitate a system: “DOS attacks try to create large streams of joins/leaves that will cause the PIM to create large routing structures or tie up the processors. If unchecked such attacks will eventually crash the system.” Ex 1005 ¶ [0072]; *see also id.* ¶¶ [0023], [0026], [0053]. Limits set on port traffic in Naik help guard against DOS attacks, and Hu correspondingly teaches techniques for setting traffic limits using a “pre-defined threshold” for a link aggregation group. These overlapping disclosures would have motivated a POSITA to combine their teachings to arrive at higher-capacity ports while still guarding against excessive traffic that, for example, may be related to a DOS attack. Ex. 1003 ¶ [0185].

A POSITA also would have been motivated to combine Naik and Hu because doing so would have entailed the use of a known technique to improve a similar device in the same way. MPEP § 2143.I(C); Ex. 1003 ¶ [0186]. Naik discloses a “base” device as an edge router R1 that transmits multicast packet traffic across separate physical ports 256 and sets a maximum limit on the number of channels over the ports. MPEP § 2143.I(C)(1). Hu presents a “comparable” device in terms of content switch/router 102. Specifically, Hu has improved switch/router 102 in that it aggregates physical ports into a link aggregation group and sets a “pre-defined threshold” for video traffic loads over the high-capacity port defined by link aggregation. MPEP § 2143.I(C)(2). A POSITA would have applied these “improvements” from Hu into the edge router R1 of Naik in the same way by following several basic technical steps. Ex. 1003 ¶ [0186]. As discussed above for element [1.6], a POSITA would have made several software modifications following IEEE 802.3ad to form a logical link between multiple physical ports as part of “**aggregating the ports.**”

Then, a POSITA would have updated software within edge router R1, such as within traffic map 506 or policy map 508, to accommodate any changes with “**setting an upper limit on a number of the multicast packet streams that are permitted for forwarding**” when conducted over a “**high-capacity port.**” *Id.* ¶ [0187]. Naik already discloses monitoring port traffic, such as the “maximum number of hosts allowed on a port, and the maximum number of groups or channels allowed on a specific port.” Ex. 1005 ¶ [0025]. Therefore, only minor modifications in software would have been made to address a “pre-defined threshold” for video traffic loads on a link aggregation group, i.e., “**over the high-capacity port,**” as taught by Hu. Ex. 1003 ¶ [0187].

Further, the results flowing from the combination of Naik and Hu would have been predictable to a POSITA and have led to a reasonable expectation of success because both references address the limiting of traffic across ports of a packet switching device. The extension of Naik to a link aggregation group as taught by Hu would be expected to expand the results that Naik obtains with using a policy map to limit traffic on physical ports to include the results that Hu obtains from using a traffic load module 116 to limit traffic on logical ports of a link aggregation group. Ex. 1003 ¶ [0188]; MPEP § 2143 at I(C)(3). Further, because Hu’s aggregation techniques follow a known standard, IEEE 802.3ad, the modifications would have been predictable. Ex. 1003 ¶ [0188].

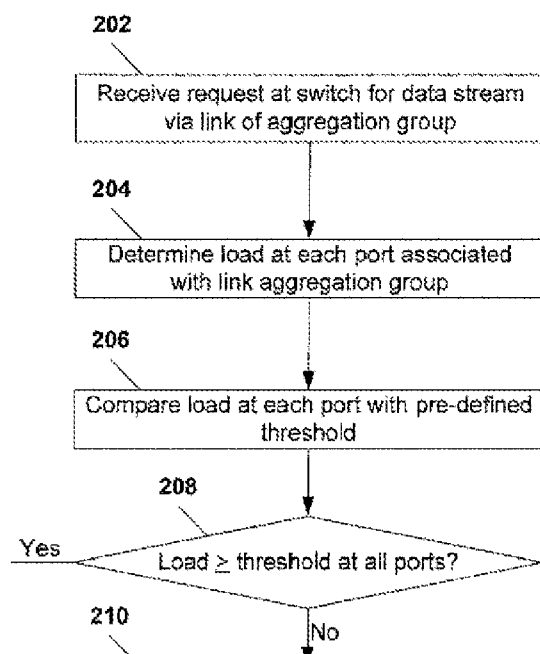
For at least these reasons, Naik and Hu disclose or suggest all limitations of element [1.7], and the combination of Naik, Reader, and Hu would have rendered obvious “**wherein aggregating the ports comprises setting an upper limit on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port,**” as recited in element [1.7]. Ex. 1003 ¶ [0189].

xiii. [1.7.1] “wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and”

Naik and Hu disclose or suggest every limitation of element [1.7.1]. Ex 1003 ¶ [0190].

Naik discloses “**wherein the request** (multicast request within IGMPv3 packets 504, such as an IGMP Join request) **indicates that the given multicast packet stream** (requested/joined multicast channel) **is to be forwarded over** a given port (sent out egress interface 430), but Naik does not disclose that the port is a “**high-capacity port.**” *Id.* Hu, however, discloses **wherein the request** (join and leave requests) **indicates that the given multicast packet stream** (joined video

stream) **is to be forwarded over the high-capacity port** (one of the links of the link aggregation group 124). *Id.*



As indicated in FIG. 2 of Hu, reproduced in part above, content switch/router 102 receives a request to join a stream via a “link of aggregation group” at block 202. Ex. 1009 at 6:11-17 (“At block 202, a switch/router system of a network, such as a multilayer content switch of a triple-play network, receives a request for a new data stream via a link of a link aggregation group.”) Hu further explains that the join request indicates the port for the video stream, which corresponds to **“wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port”** recited in element [1.7.1]:

Data related to join and leave requests can include a video stream joined or left, a join or leave time, a bit rate associated with the video stream, **an identification of a port affected by the request (e.g., a port over which a joined video stream is sent or a port over which a left video stream is no longer sent)**, or any combination thereof.

Id. at 4:3-13. Ex. 1003 ¶ [0191].

Accordingly, in combining Naik and Hu to implement link aggregation for ports 256 in edge router R1 of Naik, it would have been obvious that **“the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port,”** as recited in element [1.7.1], in order for the traffic to be processed by the router for the high-capacity port. Ex. 1003 ¶

[0192]. Otherwise, the router would not be able to properly handle routing of multicast traffic across aggregated, or high-capacity, ports. *Id.* For the same reasons discussed regarding element 1.7, a POSITA would have included Hu's teachings regarding link aggregation in implementing the combination of Naik, Reader, and Hu, which would have resulted in the understanding by a POSITA that the request can include data indicating that the requested multicast packet stream is to be forwarded over a high-capacity port, or link aggregation group, as taught by Hu. *Id.*

For at least these reasons, Naik and Hu disclose or suggest all limitations of element [1.7.1], and Naik and Hu would have rendered obvious **“the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port.”** Ex. 1003 ¶ [0193].

- xiv. [1.7.2] **“wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.”**

The combination of Naik, Reader, and Hu discloses or suggests the subject matter of element [1.7.2]. Ex. 1003 ¶ [0194].

Naik. Naik, alone or in further view of Reader, discloses all of element [1.7.2] except forwarding **“over the high-capacity port.”** Specifically, Naik discloses **wherein updating the list** (MFIB Forwarding Tables 400 within line cards 260) **comprises modifying the list** (MFIB Forwarding Tables 400 are “populated” from MRIB) **to indicate that the given multicast packet stream** (collective entries 410 in MFIB Forwarding List 400 in FIG. 4) **is permitted for forwarding over...** [a physical port]... **only when the number of the multicast packet streams that are permitted for forwarding** (tracking table 514, number of channels joined from each port, [0082]) **over ...** [a physical port],... **including the given multicast packet stream, does not exceed the upper limit** (maximum number of “joins” allowed by the switch for “LIMIT” action; deny admission 510). *Id.*

For **“modifying the list to indicate the given multicast packet stream is permitted for forwarding,”** Naik teaches to accept a multicast request, build MRIB 238 with routing information, and to derive MFIB forwarding tables 400 within line cards 260 from the MRIB. Ex. 1005 ¶ [0045]. Specifically, as noted above for element [1.4.1], Naik teaches that its forwarding tables 400 within MFIB 480, shown in FIG. 2 within each line card 260, are “populated” with

routing information from MRIB 238 within memory 230 about how to forward data packets. Ex. 1005 at FIG. 4, ¶ [0045] (the MFIB derived from the MRIB “is embodied as one or more multicast forwarding tables 400 that are populated on the line cards 260 and whose contents describe how to forward data packets through the router.”); *see supra* Section IV.A.1.a.ix (element [1.4.2]).

Reader also teaches “**modifying the list to indicate the given multicast packet stream is permitted for forwarding.**” According to Reader, in response to updates to master multicast forwarding database (MM-FDB) 350 in switch 150, switch manager 250 distributes the contents of MM-FDB 350 to each switch interface. Ex. 1008 ¶ [0021]. Slave multicast forwarding databases (SM-FDBs) (“**the list**”) in all interfaces are then commonly “maintain[ed]” (“**modif[ied]**”) to indicate “as entries multicast groups and associated switch ports through which destination hosts that are registered in the multicast groups are reachable” (“**to indicate the given multicast packet stream is permitted for forwarding**”). *Id.*

A POSITA would have understood that the populating of the forwarding tables 400 in Naik and the maintaining of slave multicast forwarding databases (SM-FDBs) in Reader indicate that “**the given multicast packet stream is permitted for forwarding**” to ensure the line cards or interfaces can properly forward the requested multicast packet stream as it arrives. Ex. 1003 ¶ [0197]. Indeed, Naik teaches: “In response to receiving the incoming packet from a neighbor, the MFIB 480 performs a lookup into its forwarding table 400 to find a route of an entry 410 that matches a destination address of the packet. The matching route instructs the router as to which interfaces the packet should be forwarded.” Ex. 1005 ¶ [0052]. And for Reader: “In IP Multicast forwarding on LAN switch 150, the SM-FDB on the ingress one of interfaces 210A through 210N, 230, 240 is invoked to resolve a multicast group address in an IP Multicast data packet to one or more switch ports, and the data packet is transmitted on all resolved switch ports, ...” Ex. 1008 ¶ [0021]. Thus, in both references “**the list**” is the authority and indicates “**the given multicast packet stream is permitted for forwarding**” to ensure correct switching operations. Ex. 1003 ¶ [0197].

Moreover, both Naik and Reader disclose controls that ensure only multicast packet streams “**permitted for forwarding**” are allowed into “**the list**,” i.e., into MFIB 480 and its forwarding table 400 in Naik and into slave multicast forwarding databases (SM-FDBs) in Reader. *Id.* ¶ [0198]. Naik includes an Admission Control Module (ACM) 500 that evaluates multicast requests and “accepts or denies these requests based on a configurable service policy.” Ex. 1005 ¶

[0023]. Similarly, Reader discloses a multicast authorization database (M-ADB) 340 that dictates whether the multicast request “is authorized to participate in the multicast group” based on a “network policy.” Ex. 1008 ¶¶ [0025], [0005]. According to Reader, “if the switch port is not authorized to participate in the multicast group, the [multicast request] packet is dropped without updating MM-FDB 350.” *Id.* ¶ [0025].

Conversely, Naik discloses that the forwarding table 400 may also be modified—by removing a requestor from a multicast group—to indicate that a multicast packet stream is *not* permitted for forwarding to that requestor. For instance, if ACM 500 concludes after applying a traffic map and a policy map that a multicast request should be denied, Naik discloses that “if already a member the requestor is removed from the group *and/or from the forwarding table.*” *Id.* ¶ [0074].

Therefore, a POSITA would understand from Naik alone or in view of Reader that the presence of a multicast packet stream within forwarding table 400 in Naik or the SM-FDBs in Reader, and the selective admission and modification of those lists, indicates that “**updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding.**” Ex. 1003 ¶ [0200].

For forwarding “**only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit,**” Naik teaches that ACM 500 accepts a multicast request—and therefore authorizes modification of the MFIB—only if the given multicast packet stream of the request would not exceed the upper limit, or maximum value, set for a port. Ex. 1003 ¶ [0201]. For instance, ACM 500 tracks the number of streams for a port in a tracking table, compares the number to a limit, and denies any further multicast join requests if the limit would be exceeded:

The ACM would track the number involved and compare it to the limit. *When the limit is reached any additional joins will always be denied* regardless of the traffic and service policy. The limit parameter may be stored in the tracking table described below, but in any event the PIM 234 and the MRIB 238 may participate in implementing such limits.

Ex. 1005 ¶ [0072]; *see also id.* ¶ [0055]. Thus, requests are accepted “**only when the number ... does not exceed the upper limit,**” as claimed. Ex. 1003 ¶ [0201]; Naik at Abstract (“Incoming packet traffic received by an edge router is snooped and, when a request to join a multicast channel

is received, the traffic is analyzed” and “the actions may be to limit the number of routing states, by denying admittance to a groups [sic] *once a limit number of requests to join, or other such parameter, is reached.*”).

Consequently, Naik alone or in view of Reader at least suggests “**wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the [physical port] only when the number of the multicast packet streams that are permitted for forwarding over the [physical port], including the given multicast packet stream, does not exceed the upper limit**” of element [1.7.2]. Ex. 1003 ¶ [0202]. Notably, the Examiner reached the same conclusion during prosecution in rejecting dependent application claim 5, which recited in part:

wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding *over a given port* only when a number of the multicast packet streams that are permitted for forwarding *over the given port*, including the given multicast packet stream, does not exceed a maximum value specified *for the given port*.

Ex. 1002 at 31. The Examiner rejected this claim based on the disclosures in paragraphs [0055] and [0072] in Naik, which are quoted from and cited above. *Id.* at 112, 192-93.

Naik does not explicitly disclose the aspect of element [1.7.2] as applied to a “**high-capacity port**.” Hu discloses this limitation. Ex. 1003 ¶ [0203].

Hu. Hu teaches the features of element [1.7.2] in the context of a high-capacity port formed from link aggregation. In particular, Hu discloses **wherein updating the list comprises modifying the list** (Request Module 118 and Traffic Load Module 116 determining real-time traffic load) **to indicate that the given multicast packet stream** (Step 202 in FIG. 2, stream requested via link of aggregation group) **is permitted for forwarding** (Policy Module 120) **over the high-capacity port** (link aggregation group 124 or 128 in FIG. 1) **only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port** (Step 204 in FIG. 2, determine load at each port), **including the given multicast packet stream** (Step 212 in FIG. 2, include bandwidth of requested stream), **does not exceed the upper limit** (Steps 214-218 in FIG. 2, assign new data stream only if video traffic load does not exceed pre-defined threshold). Ex. 1009 at 6:10-7:8; Ex. 1003 ¶ [0203].

Hu teaches to accept a multicast request only if the given multicast packet stream of the multicast request would not exceed a pre-defined threshold, or “**upper limit**,” of traffic load set

for a “**high-capacity port**” within a link aggregation group. Ex. 1003 ¶ [0204]. As stated in Hu, “[A] request can be rejected if [video traffic load on] each port of a plurality of physical ports associated with a link aggregation group is equal to or exceeds a threshold.” *Id.* at 5:60-62. This analyzed traffic load is “**including the given multicast packet stream**,” as recited in element [1.7.2]: “The request module 118 can be executable by the processing logic 104 to determine whether a total video traffic load, including the current video traffic load *and a traffic load associated with the requested video stream*, would exceed the threshold.” *Id.* at 5:34-39.

Hu also explains that “the total traffic load on a port” associated with a link aggregation group may be estimated “based on a number of join and leave requests associated with the [IGMP] group,” which corresponds to the claimed “**number of the multicast packet streams that are permitted for forwarding**.” *Id.* at 13:8-10; Ex. 1003 ¶ [0205]. Hu further explains accepting entry of a new stream request if the traffic load (e.g., “a number join and leave requests”) does not exceed a pre-defined threshold with respect to FIG. 2, which corresponds with a given “**multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit**” as claimed:

Continuing to decision node 208, the switch/router system determines whether the traffic loads at all of the ports associated with the link aggregation group meet or exceed the pre-defined threshold. *If the traffic loads at all of the ports associated with the link aggregation group meet or exceed the pre-defined threshold*, the method advances to decision node 216. ... Proceeding to decision node 216, the switch/router system determines whether an existing data stream can be dropped at any of the ports associated with the link aggregation group, such that a total traffic load at the port at which the existing data stream is dropped is lower than the pre-defined threshold. ... *[I]f such a data stream cannot be dropped, the method moves to block 218, and the request is rejected by the switch/router system.*

Id. at 6:41-7:8; Ex. 1003 ¶ [0205]. Accordingly, Hu teaches to accept a join request “**only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit**,” as recited in element [1.7.2]. Ex. 1003 ¶ [0205].

It would have been obvious in light of Hu to have modified MFIB 480 and its forwarding tables 400 in Naik “**to indicate that the given multicast packet stream is permitted for**

forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit,” as taught by Hu and recited in element [1.7.2]. *Id.* ¶ [0206].

Motivation to Combine. The previous section for element [1.7] explains why it would have been obvious to a POSITA to have combined Naik and Hu with respect to link aggregation and setting an upper limit on a number of the multicast packet streams. *See supra* Section IV.A.1.a.xii. Additionally, a POSITA would have been motivated to extend the teachings of Naik (alone or as modified by Reader) to update the forwarding tables to indicate permission for **“forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit,”** as claimed. Ex. 1003 ¶ [0207]. Enforcement of this upper limit would have helped achieve one of Naik’s goals—blocking Denial of Service attacks—even when aggregated ports are used. *Id.*; Ex. 1005 at Abstract (“[T]o limit the number of routing states, by denying admittance to a group[] once a limit number of requests to join, or other such parameter, is reached ... will substantially prevent DOS attacks on a multicast router.”); Ex. 1005 ¶¶ [0023], [0053], [0072]. As explained in Naik, DOS attacks can be catastrophic: “DOS attacks try to create large streams of joins/leaves that will cause the PIM to create large routing structures or tie up the processors. If unchecked such attacks will eventually crash the system.” *Id.* ¶ [0072].

As with the discussion with respect to element [1.7] above, combining Hu’s teachings to reject a stream request if traffic load exceeds a pre-defined threshold for a high-capacity port with the teachings of Naik regarding the usefulness of using upper limits to deter DOS attacks would have entailed the use of a known technique to improve a similar device in the same way. *See supra* Section IV.A.1.a.xii; MPEP § 2143.I(C). Under that analysis, Naik discloses a “base” device as an edge router R1 that is configured with a traffic map to block excess multicast traffic on a port that may be attributable to a DOS attack. Ex. 1003 ¶ [0208]. Hu presents a “comparable” device in content switch/router 102 that has been improved to handle high-capacity ports formed through link aggregation and to set limits through a policy module 120 on the amount of video traffic load permitted on ports within a link aggregation group. Following the “pre-defined threshold” for the video traffic loads in Hu, bandwidth on the link aggregation group can be kept within an acceptable

amount, which would extend the concepts of Naik to a high-capacity and higher bandwidth environment. *Id.*

Further, a POSITA would have had a reasonable expectation of success in making the combination, as Hu's improvements would be applied to the edge router R1 of Naik in the same way with relatively basic experimentation. Ex. 1003 ¶ [0209]. At a minimum, a POSITA would need to make little, if any, modification to Naik with respect to the subject matter of element [1.7.2] because Naik already employs traffic map 506 and policy map 508 to track the number of streams for the ports on edge router R1 and the number of users per port and accepts new join requests only if limits under the policy are not exceeded. *Id.*; see Ex. 1005 ¶¶ [0066], [0072]. Applying these techniques to logical ports joined as a link aggregation group, i.e., "**over the high-capacity port**," may require minor adjustments to software executing traffic map 506 and policy map 508. Ex. 1003 ¶ [0209].

Additionally, a POSITA would have been motivated to incorporate some of the traffic load measurements taught by Hu for aggregated ports, such as "an average bit rate," a join or leave time, "average traffic load over a period of time," "bandwidth consumption," and "a ratio of streams to traffic load." Ex. 1009 at 13:8-10, 6:26-32, 7:48-52, 4:3-7, 3:45-49, 5:22-29; Ex. 1003 ¶ [0210]. To implement these enhancements, a POSITA would have had to update software within edge router R1 of Naik, such as with traffic map 506 or policy map 508, to adjust to the types of information processed to determine whether a bandwidth passing "**over the high-capacity port**" exceeds a set limit, as taught by Hu. Ex. 1003 ¶ [0210]. These software modifications to Naik would have involved minor levels of experimentation for a POSITA. And given the overlapping disclosures of Naik and Hu relating to restricting acceptance of multicast requests based on an upper traffic limit, the modifications would have yielded predictable results consistent with what Naik already provides, including the expected ability to limit traffic load on a link aggregation group and to guard against DOS attacks. *Id.*

Moreover, Hu teaches that existing data streams in a link aggregation group may be dropped to provide bandwidth for the new data stream. See, e.g., Ex. 1009 at 5:44-56. This would provide adaptability to the network element in Naik. A POSITA would have been motivated to introduce these and other concepts of load balancing into the edge router in Naik to expand its usefulness for high-bandwidth environments. Ex. 1003 ¶ [0211].

For at least these reasons, Naik and Hu disclose or suggest all limitations of element [1.7.2], and Naik and Hu would have rendered obvious “**wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit,**” as recited in element [1.7.2]. Ex. 1003 ¶ [0212].

In summary, Naik in view of Reader, further in view of Hu, would have rendered independent claim 1 unpatentable under 35 U.S.C. § 103(a) (pre-AIA). *Id.* ¶ [0213].

b. Dependent Claims 2-7

Claims 2-7 depend from independent claim 1 and recite the additional elements quoted in the sections below. Naik in view of Reader, further in view of Hu, renders unpatentable independent claim 1 for the reasons discussed above. *See supra* Section IV.A.1.a. Moreover, the combination of Naik, Reader, and Hu discloses or suggests the additional elements of claims 2-7 as discussed below, rendering those claims unpatentable under 35 U.S.C. § 103(a) (pre-AIA). Ex. 1003 ¶ [0214].

i. 2. “The method according to claim 1, wherein storing the list, distributing the request and selectively forwarding the multicast packets are performed by dedicated hardware logic.”

Each of Naik, Reader, and Hu discloses the subject matter of claim 2. Ex. 1003 ¶ [0215].

Naik. Naik discloses **wherein storing the list** (MFIB 480 and MFIB Forwarding List 400), **distributing the request** (multicast request to source of multicast or to route processors 210 for MRIB populating MFIB 480 and MFIB Forwarding List 400) **and selectively forwarding the multicast packets** (Apply Action 714 and Track/Log IBMPv3 Traffic 716) **are performed by dedicated hardware logic** (edge router R1, two central processing units, e.g., Power-PC 7460 chips, configured as a symmetric multiprocessing (SMP) pair, route processors 210, memory 230, line cards 260, receive forwarding processors FP Rx 280, transmit forwarding processors FP Tx 290, interface 270). *Id.*

Reader. To the extent Naik is viewed as not explicitly disclosing “dedicated hardware logic” for executing the claimed functions, Reader discloses that subject matter. Reader discloses **wherein storing the list** (slave multicast forwarding databases (SM-FDBs)), **distributing the**

request (distributing contents of MM-FDB 350 to interfaces 210A-210N, 230, 240) **and selectively forwarding the multicast packets** (the SM-FDB on the ingress is invoked and data packet is transmitted on all resolved switch ports) **are performed by dedicated hardware logic** (discrete logic, such as application specific integrated circuits (ASICs) and a combination of discrete logic and programmable logic, such as programmable network processors, within interfaces 210A-N, 230, 240). Ex. 1003 ¶ [0216]. Specifically, Reader states: “Transceivers and media access controllers may be *implemented using discrete logic, such as application specific integrated circuits (ASICs), whereas packet switching engines may be implemented using a combination of discrete logic and programmable logic*, such as programmable network processors.” Ex. 1008 ¶ [0018].

Hu. To the extent Naik and Reader are viewed as not explicitly disclosing “dedicated hardware logic” for executing the claimed functions, Hu discloses that subject matter. In particular, Hu expressly discloses “**dedicated hardware logic**” as claimed for routing multicast packet streams: “[D]edicated hardware implementations, such as application specific integrated circuits, programmable logic arrays and other hardware devices, can be constructed to implement one or more of the methods described herein.” Ex. 1009 at 10:16-20; *id.* at 3:7-10 (“content switch 102 can include processing logic 104, such as one or more independent or redundant processors, and can include a memory 106 that is accessible to the processing logic 104.”).

A POSITA would have been motivated to apply the teachings of Hu (or alternatively Naik or Reader) in developing an edge router R1 as disclosed in Naik to have “dedicated hardware logic” with a reasonable expectation of success at least because doing so could have resulted in certain advantages possibly available with that type of implementation, such as smaller size, lower power consumption, and higher speed for the router. Ex. 1003 ¶ [0218].

Accordingly, each of the references cited for claim 1, together or separately, disclose or suggest “**wherein storing the list, distributing the request and selectively forwarding the multicast packets are performed by dedicated hardware logic**,” as recited in claim 2. Thus, it would have been obvious to a POSITA from the combination of Naik, Reader, and Hu to have implemented the subject matter of dependent claim 2 “**by dedicated hardware logic**.” Ex. 1003 ¶ [0219].

- ii. **3. [3.1] “The method according to claim 1, wherein distributing the request comprises duplicating the request to produce first and second copies, forwarding the first copy to a source of the given multicast packet stream, and”**

Naik in view of Reader discloses or suggests the subject matter of element [3.1]. Ex. 1003 ¶ [0220].

Naik. Naik teaches a portion of element [3.1], namely, **distributing the request comprises ... forwarding the first copy to a source of the given multicast packet stream** (edge router R1 forwards a multicast request to join to its neighboring routers on the link in the direction of the rendezvous point or multicast source). *Id.*

As explained in Section IV.A.viii.(a) above for element [1.4.1], Naik discloses that a router receiving a multicast request “communicates that request (“**forwarding**”) to its neighboring routers (neighbors) on the link towards the rendezvous point (for a shared tree) or source for a SPT [shortest path tree] (“**to a source of the given multicast packet stream**”),” from which “the routers construct a multicast distribution tree rooted at a rendezvous point or source for that group address.” Ex. 1005 ¶ [0012]. Moreover, forwarding of a received multicast request upstream to a source is a basic step of IGMP multicast routing, as a POSITA would understand, to communicate a new user or host of a multicast group to a source of the multicast packet stream. Ex. 1003 ¶ [0221].

Reader. Naik does not explicitly disclose that distributing the request comprises “**duplicating the request to produce first and second copies,**” as recited in element [3.1]. It would have been obvious to a POSITA, however, to “**duplicat[e] the request to produce first and second copies**” in view of the teachings of Reader before “**forwarding the first copy to a source of the given multicast packet stream,**” as taught by Naik. Ex. 1003 ¶ [0222].

Reader teaches that **duplicating the request** (make copy of received packets with MAC addresses not in slave forwarding database) at a receiving interface and **forwarding the first copy** (transmit copy from interface to switch manager), as recited in element [3.1].

In Reader, the interfaces 210A-210N, 230, and 240 of LAN switch 150 “snoop” incoming packets to detect if a source Media Access Control (MAC) address in the packet is not already in a local slave forwarding database. Ex. 1008 ¶ [0020]. If the MAC address is not present, “notification may be accomplished, for example, by transmitting to switch manager 250 **a copy of**

such data packets [“**duplicating the request to produce first [and second] copies**”] along with an identifier of the ingress switch port.” *Id.*

Further, Reader discloses “**forwarding [the request] to a source (router 130) of the given multicast packet stream,**” as recited in element [3.1]. Ex. 1003 ¶ [0224]. Namely, as explained in Reader, LAN switch 150 “receives an IGMP membership report from the end station (450) and determines whether the end station is authorized to join the multicast group identified in the report by reference to” the authorization database 340. Ex. 1008 ¶ [0029]. “If the host is authorized, LAN switch updates MM-FDB350 to include the new group/port association and relays (“**forwarding**”) the report (“**the request**”) to router 130 (480) (“**a source of the given multicast packet stream**”).” *Id.*; *see also id.* at FIG. 4 (step 480, stating “RELAY REPORT TO ROUTER”).

While Reader does not expressly disclose duplicating the request to produce first “**and second copies,**” duplicating multicast packets in any number, including multicast requests, was one of various options in the field for distributing the information within the packet. Ex. 1003 ¶ [0225]. These options also included forwarding the original packet, generating a new packet containing at least the information of the multicast request, and communicating the content of the request in a different form. *Id.* It would have been a matter of choice and implementation for a POSITA to have communicated information of a multicast request by duplicating and distributing multiple copies of the packet (such as “**first and second copies**”), with that choice having a reasonable expectation of success equivalent to other options for distributing the information carried by a received multicast request. *Id.*

iii. [3.2] “**distributing the second copy among the multiple processing units.**”

Naik in view of Reader discloses or suggests the subject matter of element [3.2]. Ex. 1003 ¶ [0226]. As explained above for element [1.4.1], Naik teaches “**distributing [the request] among the multiple processing units,**” as recited for element [3.2]. *See supra* Section IV.A.1.a.viii, which is incorporated by reference.

While Naik alone does not expressly disclose distributing a “**second copy**” of a received request among the multiple processing units, that variation would have been obvious to a POSITA in view of Naik. Ex. 1003 ¶ [0227]. As discussed above for element [3.1], duplicating packets into any quantity for ease of distribution within a router or switch would have been a trivial step for a POSITA and was a routine practice in the field. *Id.* Indeed, copying and distributing packets is a

basic step within a communications network to distribute packets from one source to multiple destinations for a multicast packet stream. *Id.* Moreover, the duplication of snooped packets within a router or switch was also known, as taught by Reader. *Id.*; Ex. 1008 ¶ [0020].

Further, Naik at least suggests that a second copy of the request may be distributed “**among the multiple processing units,**” as claimed. As taught by Naik, the router communicates a received multicast request “to its neighboring routers (neighbors) on the link towards the rendezvous point (for a shared tree) or source for a SPT [shortest path tree] (“**a first copy of the request**”)),” and the Multicast Route Information Base (MRIB), operating in route processors 210, “populates” the MFIB 480 and their forwarding tables 400 within each line card 260 with forwarding information from the request. Ex. 1005 ¶¶ [0012], [0045], FIG. 4. Specifically, Naik explains that the MFIB is derived from the MRIB and “is embodied as one or more multicast forwarding tables 400 that are populated (“**the second copy of the request**”) on the line cards 260 and whose contents describe how to forward data packets through the router.” *Id.* ¶ [0045]. This populating from route processors 210 to line cards 260 is necessary for the line cards 260 to properly forward the multicast packet stream that is the subject of the “request,” and corresponds to “**distributing the second copy among the multiple processing units,**” as claimed. Ex. 1003 ¶ [0228].

Reader similarly suggests “**distributing the second copy among the multiple processing units,**” as recited in element [3.2]. As discussed for element [3.1] above, after detecting that an incoming packet has a source MAC address not in a slave forwarding database, an interface in Reader sends “a copy of such data packets along with an identifier of the ingress switch port” to switch manager 250. Ex. 1008 ¶ [0020]. Thereafter, switch manager 250 “distributes the contents of MM-FDB 350 (“**the second copy**”) to interfaces 210A through 210N, 230, 240 (“**among the multiple processing units**”) in response to updates for MM-FDB 350 (“**the request**”) and thereby maintains slave multicast forwarding databases (SM-FDBs) on interfaces 210A through 210N, 230, 240.” *Id.* ¶ [0021]; Ex. 1003 ¶ [0229]. As with Naik discussed above, it would have been an obvious matter of implementation for a POSITA to choose a format to distribute information of a received join request **among the multiple processing units**, whether as the original request, as a copy of the original request, or as a different message containing and conveying the information of the request. Ex. 1003 ¶ [0229].

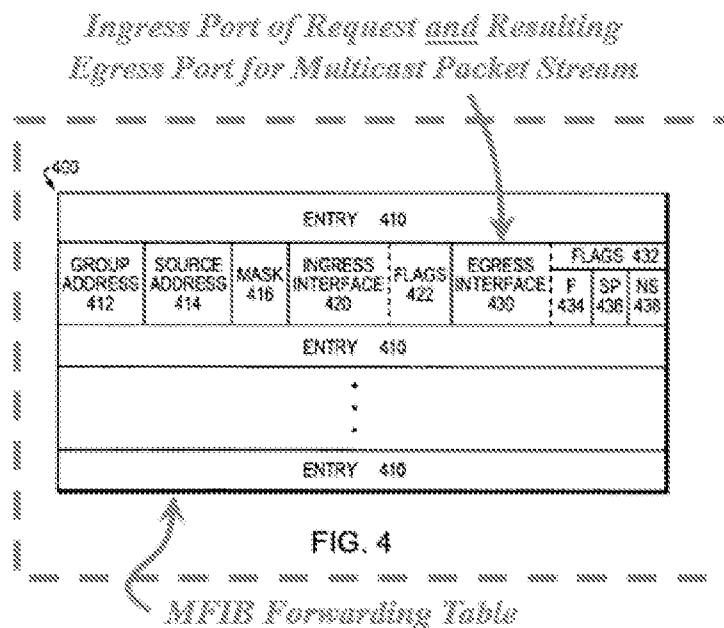
Accordingly, Naik in view of Reader discloses or suggests “**wherein distributing the request comprises duplicating the request to produce first and second copies, forwarding the first copy to a source of the given multicast packet stream, and distributing the second copy among the multiple processing units,**” and the combination of Naik, Reader, and Hu would have rendered unpatentable the subject matter of dependent claim 3. Ex. 1003 ¶ [0230].

- iv. **4. [4.1] “The method according to claim 1, wherein distributing the request comprises distributing among the multiple processing units an identity of an ingress port over which the request was received at the network element, and”**

Naik, alone or in view of Reader, discloses or suggests the subject matter of element [4.1]. Namely, Naik teaches **wherein distributing the request** (edge router R1 forwards a received multicast request to its neighboring routers on the link or after ACM 500 extracts information from the multicast request, Multicast Route Information Base (MRIB) populates MFIB 480) **comprises distributing among the multiple processing units** (line cards 260) **an identity of an ingress port** (interface 430 in MFIB Forwarding Table 400 indicating ports where packet stream exits, which includes where the multicast request was received) **over which the request** (the multicast request, such as an IGMP Join request) **was received at the network element** (edge router R1). Ex. 1003 ¶ [0231].

In Naik, ACM 500 snoops “the incoming IGMP packets and extracts certain information, for example ... the input interface (i.e., “**an identity of an ingress port over which the request was received at the network element**”). Ex. 1005 ¶¶ [0055], [0025], [0068]. Because a network switch, such as edge router R1 in Naik, forwards IGMP multicast streams to at least the port where the request was received, this identity of *an ingress port for the request* is communicated from ACM 500 in terms of an *egress port for the packet stream*. Ex. 1003 ¶ [0232]. ACM 500, which executes on route processors 210, distributes the “**ingress port over which the request was received**” to line cards 260 at least because the identity of the *egress interface for the multicast packet stream* appears within MFIB Forwarding Table 400 of MFIB 480, corresponding to “**distributing [the identity] among the multiple processing units,**” as claimed. *See* Ex. 1005 ¶¶ [0045], [0050]–[0051], FIG. 4; Ex. 1002 at 192. Specifically, as shown in annotated FIG. 4 below, MFIB Forwarding Table 400 includes an *egress interface* field 430, which according to Naik indicates the port “over which the incoming packet should be forwarded” (*id.* ¶ [0051]), and

therefore also indicates the “**identity of an ingress port over which the request was received at the network element**” (Ex. 1003 ¶ [0232]):



As stated in Naik: “[W]ith the ACM, the video packets are directed *only to the one port through which the request was received.*” *Id.* ¶ [0066].

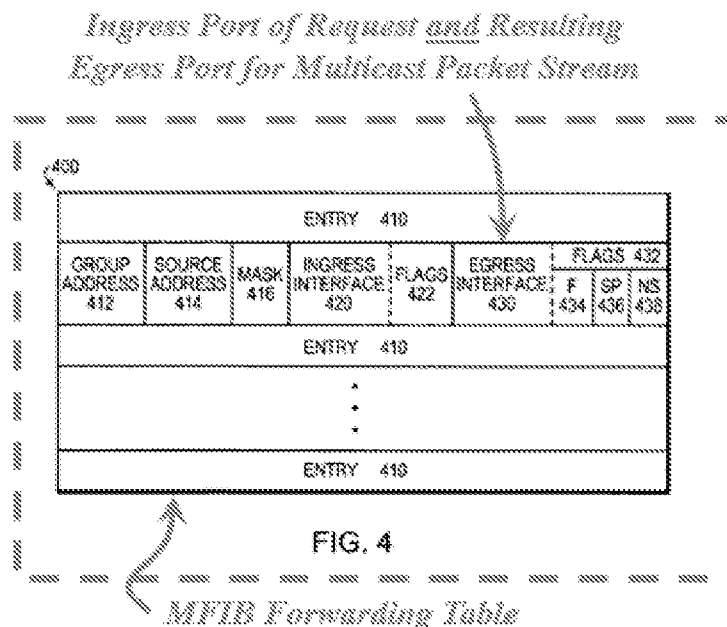
Reader further discloses “**wherein distributing the request comprises distributing among the multiple processing units an identity of an ingress port over which the request was received.**” In Reader, “LAN switch 150 ‘snoops’ IGMP packets to maintain MM-FDB 350 [master multicast forwarding database].” Ex. 1008 ¶ [0025]. “If the packet is an IGMP membership report [i.e., “**the request**”], the packet is transmitted to switch manager 250 *with an identifier of the ingress switch port* [i.e., “**an identity of an ingress port over which the request was received**”].” *Id.* If the multicast authorization database (M-ADB) 340 indicates “the switch port is authorized to participate in the multicast group,” the LAN switch “updates MM-FDB 350 to include the new multicast group/port association” and then “distributes the contents of MM-FDB 350 [“**distributing the request**”] to interfaces 210A through 210N, 230, 240 [“**among the multiple processing units**”] in response to updates to MM-FDB 350 and thereby maintains slave multicast forwarding databases (SM-FDBs) on interfaces 210A through 210N, 230, 240.” *Id.* ¶¶ [0025], [0021]; Ex. 1003 ¶ [0233]. The master database MM-FDB 350 is updated, and therefore updates the slave databases SM-FDBs, to include “as entries multicast groups and associated switch ports through which destination hosts that are registered in the multicast groups are

reachable.” *Id.* ¶ [0021]. The “new multicast group/port association” that includes “associated switch ports through which destination hosts ... are reachable” includes the “identifier of the ingress switch port” (i.e., “**an identity of an ingress port over which the request was received**”). *Id.* ¶ [0004] (“The [IGMP membership] report specifies a multicast group corresponding to the IP Multicast data stream to be joined. The LAN switch ‘snoops’ the report and *associates the group with the switch port on which the report arrived* [i.e., “**an identity of an ingress port over which the request was received**”] ...”). Ex. 1003 ¶ [0233]

Therefore, Naik, alone or in view of Reader, discloses or suggests “**wherein distributing the request comprises distributing among the multiple processing units an identity of an ingress port over which the request was received at the network element,**” as recited in element [4.1]. *Id.* ¶ [0234].

- v. [4.2] “**wherein updating the list comprises causing the list to indicate that the given multicast packet stream is permitted for forwarding over the ingress port.**”

Naik, alone or further in view of Reader, also discloses or suggests that **updating the list** (MFIB forwarding list 400) **comprises causing the list to indicate** (FIG. 4) **that the given multicast packet stream** (collective entries 410 in MFIB Forwarding List 400 in FIG. 4) **is permitted for forwarding** (present in MFIB Forwarding List 400) **over the ingress port** of the request (interface field 430 in MFIB Forwarding Table 400 indicating egress port of packet stream and, therefore, ingress port of multicast join request), as recited in element [4.2]. Ex. 1003 ¶ [0235].



As shown above in FIG. 4 (annotated outside dashed frame), an MFIB Forwarding Table 400 (“**the list**”), which lists packets that are “**permitted for forwarding**,” as claimed, includes egress interface field 430 for the multicast packet stream. Because a network switch, such as edge router R1 in Naik, forwards IGMP multicast streams to at least the port where the request was received, this identity of *an ingress port for the request* is communicated from ACM 500 in terms of an *egress port for the packet stream*. Ex. 1003 ¶ [0236]. According to Naik, egress interface 430 indicates the port “over which the incoming packet should be forwarded” as part of the multicast packet stream. Ex. 1005 ¶ [0051]. The port “over which the incoming packet should be forwarded” includes the port through which the IGMP request was received, i.e., the “**ingress port over which the request was received**.” Ex. 1003 ¶ [0236]; *id.* ¶ [0066] (“[W]ith the ACM, the video packets are directed only to the one port through which the request was received.”). Thus, the presence of an entry 410 in MFIB Forwarding Table 400 with data in egress interface 430 (indicating *egress* for multicast data stream and *ingress* for the multicast request to join the stream) corresponds to “**causing the list to indicate that the given multicast packet stream is permitted for forwarding over the ingress port**,” as recited in element [4.2]. Ex. 1003 ¶ [0236].

Reader further discloses “**wherein updating the list comprises causing the list to indicate that the given multicast packet stream is permitted for forwarding over the ingress port**.” As explained for element [1.4.2], Reader discloses “**updating the list**” at least because once authorization database M-ADB 340 “determines whether the switch port is authorized to join the multicast group identified in the [IGMP membership report],” switch manager 250 “updates MM-FDB 350 to include the new multicast group/port association” and then “distributes the contents of MM-FDB 350” to all interfaces “and thereby maintains” SM-FDBs on those interfaces. Ex. 1008 ¶¶ [0025], [0021]; *see supra* Section IV.A.1.a.ix. Reader additionally teaches that the updating causes “**the list to indicate that the given multicast packet stream is permitted for forwarding over the ingress port**” at least because of the update to MM-FDB 350, which therefore updates SM-FDBs. As a result, these two matching databases include “as entries multicast groups *and associated switch ports* through which destination hosts that are registered in the multicast groups are reachable [i.e., “**indicate that the given multicast packet stream is permitted for forwarding over the ingress port**”].” *Id.* ¶ [0021]; *see also id.* ¶ [0004] (“The [IGMP membership] report specifies a multicast group corresponding to the IP Multicast data stream to be joined. The LAN switch ‘snoops’ the report and *associates the group with the switch*

port on which the report arrived to enable forwarding of traffic addressed to the group on the switch port.” [i.e., “**indicate that the given multicast packet stream is permitted for forwarding over the ingress port**”] ...”); Ex. 1003 ¶ [0237].

As a result, Naik, alone or in view of Reader, discloses or suggests “**wherein updating the list comprises causing the list to indicate that the given multicast packet stream is permitted for forwarding over the ingress port,**” as recited in element [4.2], and the combination of Naik, Reader, and Hu would have rendered unpatentable the subject matter of dependent claim 4. Ex. 1003 ¶ [0238].

vi. **5. [5.1] “The method according to claim 1, wherein the ports forming the high-capacity port are assigned to at least two of the processing units, and”**

Naik and Hu disclose or suggest the subject matter of element [5.1].

Naik. Naik discloses **wherein the ports** (physical ports 256) ... **are assigned to ... the processing units** (ports 256 respectively assigned one of line cards 260). Naik does not disclose that the ports **forming the high-capacity port are assigned to at least two of the processing units**, as claimed. Ex. 1003 ¶ [0239].

Hu. Hu teaches **wherein the ports** (physical ports 122) **forming the high-capacity port** (Link Aggregation Group 124) **are assigned to at least two of the processing units** (processing logic assigned to ports 122).

Hu teaches that “multiple physical links of a switch/router [can] be combined into a logical port that provides much higher bandwidth” (Ex. 1009 at 1:31-33), corresponding to “**ports forming the high-capacity port.**” Hu discloses that processing logic 104, “such as one or more independent or redundant processors” (*id.* at 3:8-9), may control the ports, corresponding to “**at least two of the processing units.**” Ex. 1003 ¶ [0241].

Hu does not expressly disclose that a link aggregation group is assigned to “**at least two of the processing units.**” However, FIG. 1 in Hu illustrates a first group of physical ports 122 forming Link Aggregation Group 124 and a second group of physical ports 126 forming Link Aggregation Group 128, and nothing in Hu indicates that physical ports 122 are all assigned to, or must be assigned to, the same processing unit. Likewise, Hu does not state that physical ports grouped as 126 are all assigned to the same processing unit. To the contrary, Hu broadly indicates that “link aggregation refers to a grouping of multiple network links (i.e., communication channels or circuits) into one logical high bandwidth link” and discloses “a logical port module 110 that

stores data ... associating the physical ports with a logical port.” Ex. 1009 at 3:19-21, 3:42-45. It would have been apparent to a POSITA that in implementing link aggregation in Hu, a logical port need not have been limited to physical ports 122 assigned to a single processing unit, but instead were assignable “**to at least two of the processing units,**” for example. Ex. 1003 ¶ [0242].

Further, it would have been obvious to a POSITA to implement link aggregation to the router of Naik by grouping physical ports 256 from different line cards 260 to form one link aggregation group. *Id.* Hu provides no limitation on the order or sequence of physical ports joined into a link aggregation group, and a POSITA would have recognized that, and been motivated by the fact that, joining physical ports from different line cards to yield a single logical high-capacity port would have provided more versatility and bandwidth to the router than limiting aggregation to ports within a single line card. *Id.* And a POSITA would have had a reasonable expectation of success in forming a high-capacity port with ports assigned to at least two of the processing unit in that forming a high-capacity port through a collection of physical ports and processing units is standardized by IEEE. *Id.*; Ex. 1009 at 1:36-39.

Therefore, Naik and Hu disclose or suggest the subject matter of element [5.1], and Naik, in view of Reader, further in view of Hu, would have rendered obvious “**wherein the ports forming the high-capacity port are assigned to at least two of the processing units,**” as recited in element [5.1]. Ex. 1003 ¶ [0244].

vii. [5.2] “**wherein updating the list comprises holding respective local copies of the number of the multicast packet streams that are permitted for forwarding over the high-capacity port,**”

Naik, Reader, and Hu disclose or suggest the subject matter of element [5.2]. Ex. 1003 ¶ [0245].

Naik. Naik discloses **wherein updating the list** (MFIB 480 and MFIB Forwarding Table 400) **comprises holding respective local copies** (within edge router R1) **of the number** (number of streams in MFIB Forwarding Table 400) **of the multicast packet streams that are permitted for forwarding** (group addresses 412 in MFIB Forwarding Table 400). *Id.*

Naik discloses that, locally within edge router R1, ACM 500 will “track the number of channels joined from a port, the number of joins received from a port, the number of ports that have subscribed to a channel, and the time-period that a port was a member of a channel.” Ex. 1005 ¶ [0066]; *see id.* ¶¶ [0026], [0071] (“[T]he ACM snoops, tracks and accumulates statistics

(LOG) in an Explicit Tracking Table, the data including, 1) the number of channels joined from each port,...), [0072]. This tracking and accumulation of statistics on the number channels on each port corresponds to the claimed **“holding respective local copies of the number of the multicast packet streams that are permitted for forwarding.”** Moreover, when route processor 210, executing ACM 500 and MRIB 238, “populates” MRIB 480 with accepted or dropped channels, the populating updates MRIB Forwarding Table 400, corresponding to **“updating the list comprises holding respective local copies of the multicast packet streams that are permitted for forwarding,”** as claimed. *See* Ex. 1005 ¶ [0045]; Ex. 1003 ¶ [0246].

Naik does not disclose the subject matter of element [5.2] for forwarding **“over the high-capacity port.”** Hu teaches this subject matter.

Hu. Hu discloses **wherein updating the list comprises holding respective local copies** (within content switch/router 102) **of the number** (a number of join and leave requests) **of the multicast packet streams that are permitted for forwarding over the high-capacity port** (total traffic load on ports in link aggregation group). Ex. 1003 ¶ [0247].

More particularly, Hu discloses that locally within switch/router 102, at least by determining “a traffic load at each physical port associate with the link aggregation group through which the request was received,” which may include “estimating the total traffic load on a port ... based on a number of join and leave requests associated with the group.” Ex. 1009 at 6:23-25, 13:8-10; *see also id.* at 8:14-24 (determining “a selected port from a plurality of ports associated with a link aggregation group of the network device ... [by] select[ing] the physical port based on a stream count associated with the port”). These steps correspond with **“holding respective local copies of the number of the multicast packet streams that are permitted for forwarding over the high-capacity port,”** as claimed. Ex. 1003 ¶ [0248].

It would have been obvious to a POSITA, applying the teachings of Hu to Naik with respect to using a high-capacity port, to update the MFIB Forwarding Table 400 (i.e., **“the list”**) to reflect **“the number of the multicast packet streams that are permitted for forwarding over the high-capacity port,”** as claimed. *Id.* ¶ [0249]. Indeed, failure to include the number of streams permitted for forwarding over the high-capacity port in MFIB Forwarding Table 400 could lead to errant behavior by edge router R1 in routing multicast packet streams and maintaining an accurate Explicit Tracking Table for reporting statistics to end users and ensuring that relevant thresholds are enforced. *See* Ex. 1005 ¶ [0085]; Ex. 1003 ¶ [0249]. Moreover, a POSITA would have had a

reasonable likelihood of success in making the modification because Naik already lists the number of multicast packet streams permitted for forwarding over ports locally in forwarding table 400 and including those streams permitted for forwarding over “the high-capacity port” would be an insignificant technical change in data storage. Ex. 1003 ¶ [0249].

Accordingly, Naik and Hu disclose or suggest “**wherein updating the list comprises holding respective local copies of the number of the multicast packet streams that are permitted for forwarding over the high-capacity port,**” as recited in element [5.2].” Ex. 1003 ¶ [0250].

viii. [5.3] “updating the local copies based on the distributed request, and”

Naik and Hu disclose or suggest the subject matter of element [5.3]. Ex. 1003 ¶ [0251].

Naik. Naik teaches **updating the local copies** (within edge router R1, the number of streams in MFIB Forwarding Table 400) **based on the distributed request** (received multicast request passed to ACM 500 and distributed to MFIB 480 via MRIB 238). *Id.*

Naik discloses that, locally within edge router R1, ACM 500 will “track the number of channels joined from a port, the number of joins received from a port, the number of ports that have subscribed to a channel,” for example, as part of collecting statistics “on a per port per user basis.” Ex. 1009 ¶¶ [0026], [0071], [0072]. “Tracking” and “collecting statistics” involves “**updating the local copies,**” as claimed, as changes occur, including the number of “the multicast packet streams that are permitted for forwarding over” the port, which is “the number of channels joined from a port” following a multicast request. Similarly, with respect to entries within MFIB Forwarding Table 400, “**updating the local copies based on the distributed request**” maps to MRIB “populating” MRIB 480 in response to a snooped multicast request being accepted. *See* Ex. 1005 ¶¶ [0045], [0052], [0054]. Ex. 1003 ¶ [0252].

Hu. Hu also teaches **updating the local copies** (within content switch/router 102) **based on the distributed request** (IGMP Join request). *Id.* ¶ [0253].

Hu discloses that locally within switch/router 102, the device may be “estimating the total traffic load on a port... based on a number of join and leave requests associated with the group” and “select[ing] the physical port based on a stream count associated with the port.” Ex. 1009 at 13:8-10, 8:14-24. Estimating a load based on “a number of join and leave requests” and selecting a port based on “a stream count” involves “**updating the local copies,**” as claimed, as changes

occur. These changes include the number of “**the multicast packet streams that are permitted for forwarding over**” the port, which is “the number of channels joined from a port” following an IGMP Join request, i.e., the “**distributed request**.” Ex. 1009 at 8:60-64 (“The network device can transmit a new media stream via a selected port from the plurality of ports based on the real-time port usage and *add an estimated bandwidth associated with the new media stream* to the real-time port usage for the selected port.”). Ex. 1003 ¶ [0253].

It would have been obvious to a POSITA, applying the teachings of Hu to Naik with respect to using a high-capacity port, to update a number of streams in the MFIB Forwarding Table 400 (i.e., “**the local copies**”) based on “**the distributed request**,” as claimed. Ex. 1003 ¶ [0254]. Indeed, failure to update the number of streams permitted for forwarding over the high-capacity port in MFIB Forwarding Table 400 when an incoming Join request is distributed could lead to errant behavior by edge router R1 in routing multicast packet streams and maintaining an accurate Explicit Tracking Table for reporting statistics to end users. *See* Ex. 1005 ¶ [0085]; Ex. 1003 ¶ [0254]. Moreover, a POSITA would have had a reasonable likelihood of success in making the modification because Naik already lists the number of multicast packet streams permitted for forwarding over ports locally in forwarding table 400 and updating those streams based on a distributed request would be an insignificant technical change in data storage. Ex. 1003 ¶ [0254].

Accordingly, Naik and Hu disclose or suggest “**updating the local copies based on the distributed request**,” as recited in element [5.3]. Ex. 1003 ¶ [0255].

ix. **[5.4] “selectively updating the list in each of the at least two processing units based on the respective local copies.”**

Naik and Hu disclose or suggest the subject matter of element [5.4].

Naik. Naik teaches **selectively updating the list** (populating MFIB 480 and MFIB Forwarding Table 400) **in each of the at least two processing units** (two of line cards 260) **based on the respective local copies** (within edge router R1, the number of streams in MFIB Forwarding Table 400). Ex. 1003 ¶ [0256].

As explained for element [1.7.2] above, “**the list**” in Naik (i.e., MFIB 480 and MFIB Forwarding List 400) is updated to accept a multicast request if “**local copies**” of “the number of the multicast packet streams that are permitted for forwarding” does not exceed a maximum value. *See supra* Section IV.A.1.a.xiv.element [1.7.2]; Ex. 1005 ¶¶ [0072], [0055], Abstract, [0045]. Therefore, “**updating the list**” in Naik is “**based on the respective local copies**,” as claimed. Ex.

1003 ¶ [0257]. Similarly, if the “**local copies**” of the number of streams would exceed the maximum, the multicast request would be denied and the requestor possibly removed, further indicating that “**updating the list**” of the number of streams is “**based on the respective local copies**,” as claimed. *See* Ex. 1005 ¶ [0072] (“If the action is to deny admission, the requestor does not become a member of the group, or if already a member **the requestor is removed from the group and/or from the forwarding table**.”). *Id.* ¶ [0074]; Ex. 1003 ¶ [0257].

Naik does not disclose selectively updating the list “in each of the at least two processing units” associated with a high-capacity port, as recited in claim 5. Hu teaches this subject matter. Ex. 1003 ¶ [0258].

Hu. Hu teaches **selectively updating the list in each of the at least two processing units** (processing logic assigned to ports 122) **based on the respective local copies** (within content switch/router 102, a number of join and leave requests). *Id.*

Hu discloses monitoring a bandwidth or video traffic load across a high-capacity port, including “estimating the total traffic load on a port ... based on a number of join and leave requests associated with the group” (Ex. 1009 at 6:23-25) and, therefore, discloses “**updating the list**” based on “**the respective local copies**” of the number of join and leave requests, as recited in element [5.4]. *Id.*

It would have been readily apparent to a POSITA, applying the teachings of Hu to Naik with respect to using a high-capacity port, to update the MFIB Forwarding Table 400 (i.e., “**the list**”) based on “**the respective local copies**” [of the number of join and leave requests] even when implementing the high-capacity port in “**at least two processing units**,” as claimed. MFIB 480 is the same in all line cards 260 (*see, e.g.*, Ex. 1005 at FIG. 2), and updating each of the processing units with routing information would be necessary for accurate forwarding of packets. Ex. 1003 ¶ [0260]. Indeed, failure to include the number of streams permitted for forwarding over the high-capacity port in MFIB Forwarding Table 400 on “**the at least two processing units**” could lead to errant behavior by edge router R1 in routing multicast packet streams and maintaining an accurate Explicit Tracking Table for reporting statistics to end users. *See* Ex. 1005 ¶ [0085]; Ex. 1003 ¶ [0260]. Moreover, a POSITA would have had a reasonable likelihood of success in making the modification because Naik already lists the number of multicast packet streams permitted for forwarding over ports locally in forwarding table 400 and making updates to those lists when

implementing the high-capacity port in “at least two processing units” would be an insignificant technical change in data storage. Ex. 1003 ¶ [0260].

Accordingly, Naik and Hu disclose or suggest “**selectively updating the list in each of the at least two processing units based on the respective local copies,**” as recited in element [5.4], and the combination of Naik, Reader, and Hu would have rendered unpatentable the subject matter of dependent claim 5. *Id.* ¶ [0261].

- x. **6. “The method according to claim 1, wherein the request specifies a requested action selected from a group of actions consisting of receiving the given multicast packet stream and stopping to receive the given multicast packet stream.”**

Naik discloses the subject matter of claim 6, i.e., **wherein the request** (multicast request, including IGMPv3 packets) **specifies a requested action selected from a group of actions consisting of receiving the given multicast packet stream** (multicast request as IGMP Join request) **and stopping to receive the given multicast packet stream** (multicast request as IGMP Leave request). Ex. 1003 ¶ [0262].

Naik discloses claim 6 in that its admission control module (ACM) performs snooping on received a received multicast request (the “**request**”), which may include join or leave messages. Ex. 1005 ¶¶ [0025], [0052], [0054]. ACM 500 monitors “incoming request[s] issued by the end nodes to join, leave or modify membership in multicast groups designated to receive data” (*id.* ¶ [0053]), corresponding to “**a requested action selected from ... receiving the given multicast packet stream,**” i.e., join the stream, or “**stopping to receive the given multicast packet stream,**” i.e., leave the stream. Ex. 1002 at 193; Ex. 1003 ¶ [0263].

Accordingly, at least Naik discloses or suggests “**wherein the request specifies a requested action selected from a group of actions consisting of receiving the given multicast packet stream and stopping to receive the given multicast packet stream,**” and the combination of Naik, Reader, and Hu would have rendered unpatentable the subject matter of dependent claim 6. Ex. 1003 ¶ [0264].

- xi. 7. “The method according to claim 1, wherein the request conforms to one of an Internet Group Management Protocol (IGMP) and a Multicast Listener Discovery (MLD) protocol.”**

Naik discloses the subject matter of claim 7. In particular, Naik discloses **wherein the request (multicast request) conforms to one of an Internet Group Management Protocol (IGMP) (IGMPv3 request) and a Multicast Listener Discovery (MLD) protocol** (“conventional Multicast Listener Discovery (MLD) protocol is substantially similar to, and performs the same functions as, IGMP, but for IPv6”). Ex. 1003 ¶ [0265].

Naik explains that the multicast request received and snooped by ACM 550 is a join request under IGMPv3, corresponding with a “**request conforms to ... an Internet Group Management Protocol (IGMP) ... protocol.**” Ex. 1005 ¶ [0054] (“The ACM 500 includes a process 502 that snoops 502 the IGMPv3 packets 504, e.g. the joins and reports from all end users, and classifies the traffic via a traffic map 506.”). Naik further explains that IGMP applies to Internet Protocol version 4 (IPv4), while Multicast Listener Discovery (MLD) protocol performs the same functions for Internet Protocol version 6 (IPv6): “Note that IGMP is an IPv4 group membership protocol; the conventional **Multicast Listener Discovery (MLD) protocol** is substantially similar to, and performs the same functions as, IGMP, but for IPv6.” *Id.* ¶ [0007].

Thus, Naik discloses or suggests “**wherein the request conforms to one of an Internet Group Management Protocol (IGMP) and a Multicast Listener Discovery (MLD) protocol,**” and the combination of Naik, Reader, and Hu would have rendered unpatentable the subject matter of dependent claim 7. Ex. 1003 ¶ [0267].

c. Independent Claim 11

- i. [11.pre] “A network element, comprising;”

Naik discloses or suggests the subject matter of element [11.pre]. *See supra* Discussion of Naik in Ground 1, [1.pre], which is incorporated by reference.

- ii. [11.1] “multiple ports; and”

Naik discloses or suggests the subject matter of element [11.1]. *See supra* Discussion of Naik in Ground 1, [1.1], which is incorporated by reference.

- iii. **[11.2] “multiple interconnected processing units, [11.2.1] wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and”**

Naik discloses or suggests the subject matter of elements [11.2], [11.2.1]. *See supra* Discussion of Naik in Ground 1, [1.1], [1.1.1], and [1.1.2], which is incorporated by reference.

- iv. **[11.2.2] “wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned,”**

Naik discloses or suggests the subject matter of element [11.2.2]. *See supra* Discussion of Naik in Ground 1, [1.1.3], which is incorporated by reference.

- v. **[11.2.3] “wherein the processing units are configured to process packets that are communicated over respective subsets of one or more of the ports,”**

Naik discloses or suggests the subject matter of element [11.2.3]. *See supra* Discussion of Naik in Ground 1, [1.1.3], which is incorporated by reference.

- vi. **[11.3] “to store respective lists,” [11.3.1] “each of which indicates all of one or more multicast packet streams that are permitted for forwarding by the network element and”**

Naik discloses or suggests the subject matter of elements [11.3], [11.3.1]. *See supra* Discussion of Naik in Ground 1, [1.2] and [1.2.1], which is incorporated by reference.

- vii. **[11.3.2] “further indicates which of the one or more ports multicast packet stream is permitted to be forwarded to,”**

Naik discloses or suggests the subject matter of element [11.3.2]. *See supra* Discussion of Naik in Ground 1, [1.2.2], which is incorporated by reference.

- viii. **[11.4] “to receive from a client a request relating to reception of a given multicast packet stream, and”**

Naik discloses or suggests the subject matter of element [11.4]. *See supra* Discussion of Naik in Ground 1, [1.3], which is incorporated by reference.

- ix. **[11.5] [11.5.1] “to distribute the request among the processing units within the network element”**

Naik discloses or suggests the subject matter of elements [11.5], [11.5.1]. *See supra* Discussion of Naik in Ground 1, [1.4] and [1.4.1], which is incorporated by reference.

- x. **[11.5.2] “so as to update the lists responsively to the distributed request and”**

Naik, or Naik in view of Reader, discloses or suggests the subject matter of element [11.5.2]. *See supra* Discussion of Naik and Reader in Ground 1, [1.4.2], which is incorporated by reference.

- xi. **[11.5.3] “to selectively forward multicast packets associated with the given multicast packet stream in accordance with the updated lists,”**

Naik or Naik in view of Reader discloses the subject matter of element [11.5.2]. *See supra* Discussion of Naik and Reader in Ground 1, [1.5], which is incorporated by reference.

- xii. **[11.6] “wherein two or more of the ports are aggregated to form a high-capacity port, and”**

Naik in view of Hu discloses or suggests the subject matter of element [11.6]. *See supra* Discussion of Naik and Hu in Ground 1, [1.6], which is incorporated by reference.

- xiii. **[11.7] “wherein an upper limit is set on a number of the multicast packet streams that are permitted for forwarding over the high-capacity port,”**

Naik and Hu disclose or suggest the subject matter of element [11.7]. *See supra* Discussion of Naik and Hu in Ground 1, [1.7], which is incorporated by reference.

- xiv. **[11.7.1] “wherein the request indicates that the given multicast packet stream is to be forwarded over the high-capacity port, and”**

Naik and Hu disclose or suggest the subject matter of element [11.7.1]. *See supra* Discussion of Naik and Hu in Ground 1, [1.7.1], which is incorporated by reference.

- xv. **[11.7.2] “wherein the processing units are configured to update the lists to indicate that the given multicast packet stream is permitted for forwarding over the high-capacity port only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit.”**

Naik and Hu disclose or suggest the subject matter of element [11.7.2]. *See supra* Discussion of Naik and Hu in Ground 1, [1.7.2], which is incorporated by reference.

d. Dependent Claims 12-15 and 17

- i. **12. “The network element according to claim 11, wherein the processing units comprise dedicated hardware logic for storing the lists, distributing the request and selectively forwarding the multicast packets.”**

Naik discloses the subject matter of claim 12. *See supra* Discussion of Naik in Ground 1, [claim 2], which is incorporated by reference.

- ii. **13. [13.1] “The network element according to claim 11, wherein a processing unit receiving the request is configured to duplicate the request to produce first and second copies, so as to forward the first copy to a source of the given multicast packet stream and”**

Naik discloses the subject matter of element [13.1]. *See supra* Discussion of Naik in Ground 1, [3.1], which is incorporated by reference.

- iii. **[13.2] “distribute the second copy among the multiple processing units.”**

Naik discloses the subject matter of element [13.2]. *See supra* Discussion of Naik in Ground 1, [3.2], which is incorporated by reference.

- iv. **14. [14.1] “The network element according to claim 11, wherein a processing unit receiving the request is configured to distribute among the multiple processing units an identity of an ingress port over which the request was received, and”**

Naik and Hu disclose the subject matter of element [14.1]. *See supra* Discussion of Naik in Ground 1, [4.1], which is incorporated by reference.

- v. **[14.2] “wherein the processing units are configured to cause the respective lists to indicate that the given multicast packet stream is permitted for forwarding over the ingress port.”**

Naik and Hu disclose the subject matter of elements [14.2]. *See supra* Discussion of Naik and Hu in Ground 1, [4.2], which is incorporated by reference.

- vi. **15. [15.1] “The network element according to claim 11, wherein the ports forming the high-capacity port are assigned to at least two of the processing units, and”**

Hu discloses the subject matter of element [15.1]. *See supra* Discussion of Hu in Ground 1, [5.1], which is incorporated by reference.

- vii. **[15.2] “wherein the at least two processing units are configured to hold respective local copies of the number of the multicast packet streams that are permitted for forwarding over the high-capacity port,”**

Naik and Hu disclose the subject matter of element [15.2]. *See supra* Discussion of Naik and Hu in Ground 1, [5.2], which is incorporated by reference.

- viii. **[15.3] “to update the local copies based on the distributed request, and”**

Naik and Hu disclose the subject matter of element [15.3]. *See supra* Discussion of Naik and Hu in Ground 1, [5.3], which is incorporated by reference.

- ix. **[15.4] “to selectively update their respective lists based on the respective local copies.”**

Naik and Hu discloses the subject matter of element [15.4]. *See supra* Discussion of Naik and Hu in Ground 1, [5.4], which is incorporated by reference.

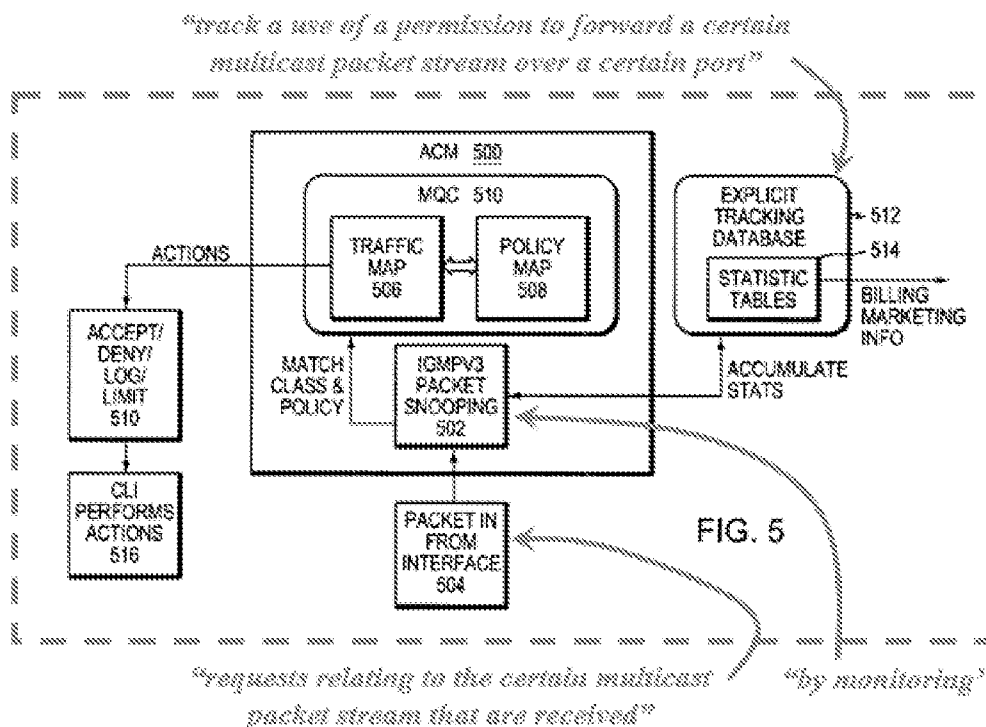
- x. **17. “The network element according to claim 11, wherein the request conforms to one of an Internet Group Management Protocol (IGMP) and a Multicast Listener Discovery (MLD) protocol.”**

Naik discloses the subject matter of claim 17. *See supra* Discussion of Naik in Ground 1, claim 7, which is incorporated by reference.

2. **Ground 2: Naik in View of Reader Further in View of Hu, Further in View of Deng, Would Have Rendered Obvious Claim 16**

- a. 16. [16.1] “The network element according to claim 11, wherein the processing units are configured to track a use of a permission to forward a certain multicast packet stream over a certain port by monitoring requests relating to the certain multicast packet stream that are received at the network element, and”

Naik discloses the subject matter of element [16.1]. Ex. 1003 ¶ [0293]. Specifically, Naik discloses **wherein the processing units** (route processors 210 or line cards 260) **are configured to track** (ACM 500 and explicit tracking database 512 in FIG. 5) **a use of a permission to forward a certain multicast packet stream** (logging how long channel is active or up or time period that a port was a member of channel) **over a certain port** (any of ports 256) **by monitoring requests relating to the certain multicast packet stream** (IGMPv3 packet snooping 502) **that are received** (incoming packets 504) **at the network element** (edge router R1). *Id.*



As shown above in annotated FIG. 5, Naik includes a functional block diagram of its edge router R1. Admission control module (ACM) 500 is executed on route processors 210 (“**the processing units**”) “to monitor incoming request[s] issued by the end nodes to join, leave or modify membership in multicast groups designated to receive data.” Ex. 1005 ¶ [0053]. Thus,

incoming IGMPv3 packets 504, shown above in FIG. 5, correspond to the claimed **“requests relating to the certain multicast packet stream that are received at the network element,”** while the snooping process 502, also shown above, corresponds to **“by monitoring [the] requests,”** as recited in element [16.1]. Ex. 1003 ¶ [0294].

Naik explains that the ACM will track activity on incoming packets 504 and record statistics within an explicit tracking database 512, corresponding to the claimed **“track a use of a permission to forward a certain multicast packet stream over a certain port.”** Ex. 1005 ¶ [0053]; Ex. 1003 ¶ [0295]. Just as in the '394 Patent where an “aging counter keeps track of a membership of a particular port in a particular multicast stream” to see which “are refreshed from time to time,” tracking by ACM 500 in Naik will include “the number of channels joined from a port, ... and the time-period that a port was a member of a channel.” Ex. 1005 ¶¶ [0066], [0071]. Naik further explains how ACM 500 will **“track a use of a permission to forward,”** as claimed: “An example of a logging record in the tracking table, 514, for each source channel, may include the following: a) *a time stamp for the first join* and the MAC address of the receiver; b) *the time stamp of the last to leave* and the MAC address of the receiver; c) *how long the channel is active or up; ...*” *Id.* ¶¶ [0075]-[0078].

Therefore, Naik discloses or suggests **“wherein the processing units are configured to track a use of a permission to forward a certain multicast packet stream over a certain port by monitoring requests relating to the certain multicast packet stream that are received at the network element,”** as recited in element [16.1]. Ex. 1003 ¶ [0296].

b. [16.2] “to remove the permission from the respective lists responsively to determining that the permission is aged.”

Naik does not expressly disclose the subject matter of element [16.2]. Deng, however, does disclose **to remove the permission** (remove multicast address) **from the respective lists** (from multicast address table) **responsively to determining that the permission is aged** (when timer 330 expires). Ex. 1003 ¶ [0297].

Deng is directed to a data link layer switch. A multicast matcher 310 stores a multicast address in a multicast address table 320 for an incoming packet “only if the packet is a request to join a multicast group.” Ex. 1007 ¶ [0035]. “In addition to storing the multicast address, the matcher 310 starts or restarts the timer 330. The timer 310 [sic, 330] is used to determine the length of time that an address remains in the multicast address table 320.” *Id.* ¶ [0036]. Thus, through

starting and restarting a time with each request to join a multicast group, Deng tracks “**a use of a permission to forward a certain multicast packet stream over a certain port**” and monitors “**requests relating to the certain multicast packet stream,**” as recited in element [16.1]. Ex. 1003 ¶ [0298].

Moreover, disclosing a timer 330 for monitoring a frequency of requests to join a multicast group, Deng teaches “**determining that the permission is aged,**” as recited in element [16.2]. Specifically, Deng states that timer 330, which matcher 310 “starts or restarts” with each request to join a multicast group (Ex. 1007 ¶ [0035]), “is used to determine the length of time that an address remains in the multicast address table 320” (*id.* ¶ [0036]), which corresponds to “**determining that the permission is aged,**” as claimed. Ex. 1003 ¶ [0299].

Further, according to Deng, “An address is removed from the table upon the expiration of the timer.” *Id.* As Deng further summarizes: “The timer determines an amount of time that the generated multicast address remains in the multicast address table.” *Id.* at Abstract. This removal upon timer expiration equals “**to remove the permission from the respective lists responsively to determining that the permission is aged,**” as recited in element [16.2]. Ex. 1003 ¶ [0300].

Motivation to Combine. A POSITA would have been motivated to implement Deng’s teachings to check for and remove old entries in a forwarding table into Naik’s edge router R1 with a reasonable expectation of success because removing entries in a forwarding table that have not been used for a long time would avoid memory clutter and avoid potential inefficiencies or errors when forwarding active multicast packet streams. Ex. 1003 ¶ [0301]. The Examiner agreed, noting that a POSITA would have combined the teachings of Deng and Naik “to support efficient multicasting by transmitting packets on only those ports with hosts that have previously joined the multicast group in the networking system.” Ex. 1002 at 208; Ex. 1003 ¶ [0301].

The references provide a teaching, suggestion, and motivation for their combination. *See* MPEP § 2143.I. For example, Naik addresses the importance of tracking activity on the ports of its network device and in blocking access to new join requests that would exceed an upper limit. Ex. 1005 at Abstract. Indeed, Naik collects statistics on each port and generates an “Explicit Tracking Database 512” with statistic tables 514 for reporting to others. *Id.* ¶¶ [0026], [0071], [0085]. Among the tracking includes “the number of channels joined from a port,” “the number of joins received from a port,” “the number of ports that have subscribed to a channel,” “the time-period that a port was a member of a channel” and “how long the channel is active or up.” *Id.* ¶¶

[0066], [0078]. Because Naik is concerned with excessive traffic on a port, in part to avoid the risk of Denial of Service attacks (*id.* ¶¶ [0023], [0072]), a POSITA would have been motivated to find ways for edge router R1 in Naik to reduce the number of channels forwarded by the router. Ex. 1003 ¶ [0302]. Deng also discloses a packet switching device for handling multicast channels and teaches that multicast channels forwarded by the device may be eliminated when they have been present on a forwarding list without activity beyond a period of time, which would have been an attractive option for a POSITA in view of Naik. Ex. 1007 ¶¶ [0035]-[0036]; Ex. 1003 ¶ [0302].

Following the teachings of Deng, a POSITA would have modified software within a network device, such as edge router R1 in Naik, to remove an entry from a forwarding list, such as MFIB 480 or forwarding table 400, when an entry has not been used beyond a period of time, according to the teachings of Deng. Ex. 1003 ¶ [0303]. As in MPEP § 2143.I(C), the implementation of a timer as taught by Deng (and implied by Naik through its measurements of a “time-period” and “how long,” Ex. 1005 ¶¶ [0066], [0078]) into a “base” device as edge router R1 in Naik from the “comparable” device of Deng would have led to predictable results for a POSITA. Ex. 1003 ¶ [0303]. In particular, the software modifications to “remove the permission from the respective lists responsively to determining that the permission is aged” would have involved insignificant software development for implementation into edge router R1 of Naik. *Id.* Naik already tracks temporal issues relating to channels and subscriber usage, so an extension of the software in Naik to modify policy map 508 to eliminate entries from forwarding table 400 after a period of time would have been insignificant to a POSITA. As a result, a POSITA would have had a reasonable expectation of success at implementing changes in software for edge router R1 in Naik that correspond with element [16.2]. *Id.*

Accordingly, Deng discloses or suggests “to remove the permission from the respective lists responsively to determining that the permission is aged,” as recited in element [16.2], and the combination of Naik, Reader, Hu, and Deng would have rendered unpatentable the subject matter of dependent claim 16. *Id.* ¶ [0304].

B. Processor-Initializing Claims

1. Ground 3: Naik in View of Reader Further in View of Ramanathan and Schuler Would Have Rendered Obvious Claims 8, 9, and 18

a. Independent Claim 8

i. [8.pre] “A method for communication, comprising:”

Naik discloses or suggests the subject matter of element [8.pre]. *See supra* Discussion of Naik in Ground 1, [1.1], which is incorporated by reference.

ii. [8.1] “operating a network element,” [8.1.1] “wherein the network element further comprises multiple interconnected processing units and multiple ports,”

Naik discloses or suggests the subject matter of element [8.1] and [8.1.1]. *See supra* Discussion of Naik in Ground 1, [1.1] and [1.1.1], which is incorporated by reference.

iii. [8.1.2] “wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and”

Naik discloses or suggests the subject matter of elements [8.1.2]. *See supra* Discussion of Naik in Ground 1, [1.1.2], which is incorporated by reference.

iv. [8.1.3] “wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned;”

Naik discloses or suggests the subject matter of element [8.1.3]. *See supra* Discussion of Naik in Ground 1, [1.1.3], which is incorporated by reference.

v. [8.2] “storing in each processing unit a respective list,” [8.2.1] “each list indicating all of one or more multicast packet streams that are permitted for forwarding by the network element, and”

Naik discloses or suggests the subject matter of elements [8.2], [8.2.1]. *See supra* Discussion of Naik in Ground 1, [1.2] and [1.2.1], which is incorporated by reference.

vi. [8.2.2] “further indicating which of the one or more ports each multicast packet stream is permitted to be forwarded to;”

Naik discloses or suggests the subject matter of element [8.2.2]. *See supra* Discussion of Naik in Ground 1, [1.2.2], which is incorporated by reference.

vii. [8.3] “receiving from a client a request relating to reception of a given multicast packet stream;”

Naik discloses or suggests the subject matter of element [8.3]. *See supra* Discussion of Naik in Ground 1, [1.3], which is incorporated by reference.

viii. [8.4] [8.4.1] “distributing the request among the multiple processing units within the network element, and”

Naik discloses or suggests the subject matter of elements [8.4], [8.4.1]. *See supra* Discussion of Naik in Ground 1, [1.4] and [1.4.1], which is incorporated by reference.

ix. [8.4.2] “updating the respective list in each processing unit responsively to the request; and”

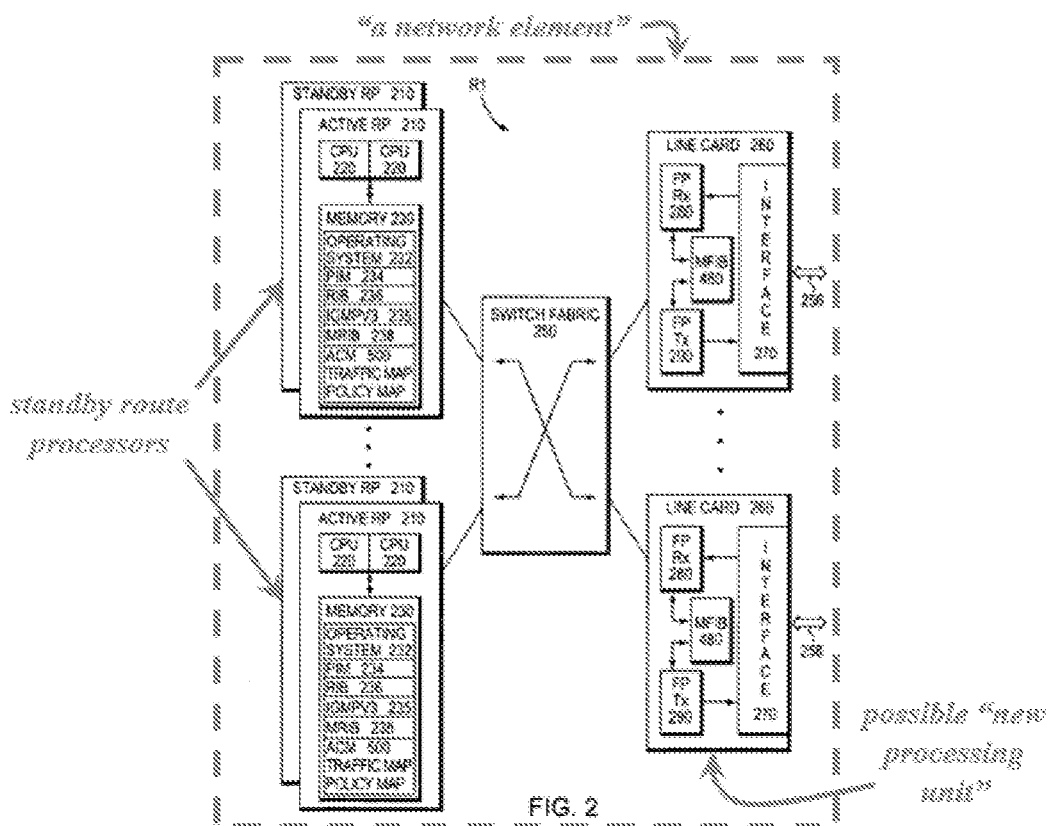
Naik, or Naik in view of Reader, discloses or suggests the subject matter of element [8.4.2]. *See supra* Discussion of Naik and Reader in Ground 1, [1.4.2], which is incorporated by reference.

x. [8.5] “selectively forwarding multicast packets associated with the given multicast packet stream by the processing units in accordance with the respective updated lists,”

Naik, or Naik in view of Reader, discloses or suggests the subject matter of element [8.5]. *See supra* Discussion of Naik and Reader in Ground 1, [1.5], which is incorporated by reference.

xi. [8.6] “wherein operating the network element comprises initializing a new processing unit added to the network element by:”

Naik discloses or suggests the subject matter of element [8.6], i.e., **wherein operating the network element** (router operating system 232 on edge router R1) **comprises initializing** (hot-swapping or bringing up a standby backup) **a new processing unit** (a new line card 260) **added to the network element** (edge router R1). Ex. 1003 ¶ [0315].



As shown above in FIG. 2 (annotated outside dashed frame), and discussed above for element [8.1], Naik discloses that edge router R1 (“**the network element**”) includes multiple route processors 210 and multiple line cards 260 interconnected through switch fabric 250. Ex. 1003 ¶¶ [0040]-[0041], [0045]. **Operating the network element**, as claimed, is done by a router operating system 232 that implements multicast components as software process modules distributed over route processors 210 and line cards 260, which are “**processing units**.” *Id.* ¶ [0045]; Ex. 1003 ¶ [0316]. According to Naik, both “active route processors 210 and line cards 260 . . . are individually hot-swappable and may have redundant (standby) backups.” *Id.* FIG. 2 expressly illustrates standby route processors 210, labeled “STANDBY RP 210,” offset from “ACTIVE RP 210,” and the ellipses between the swappable line cards 260 suggest that “redundant (standby) backups” are available for those devices. Ex. 1003 ¶ [0316]. Because hot-swapping or bringing online a redundant (standby) backup for one of route processors 210 or line cards 260 would require initializing that new processing unit into the active operating environment, Naik discloses or suggests that operating the network element may involve the claimed “**initializing a new processing unit added to the network element.**” *Id.*

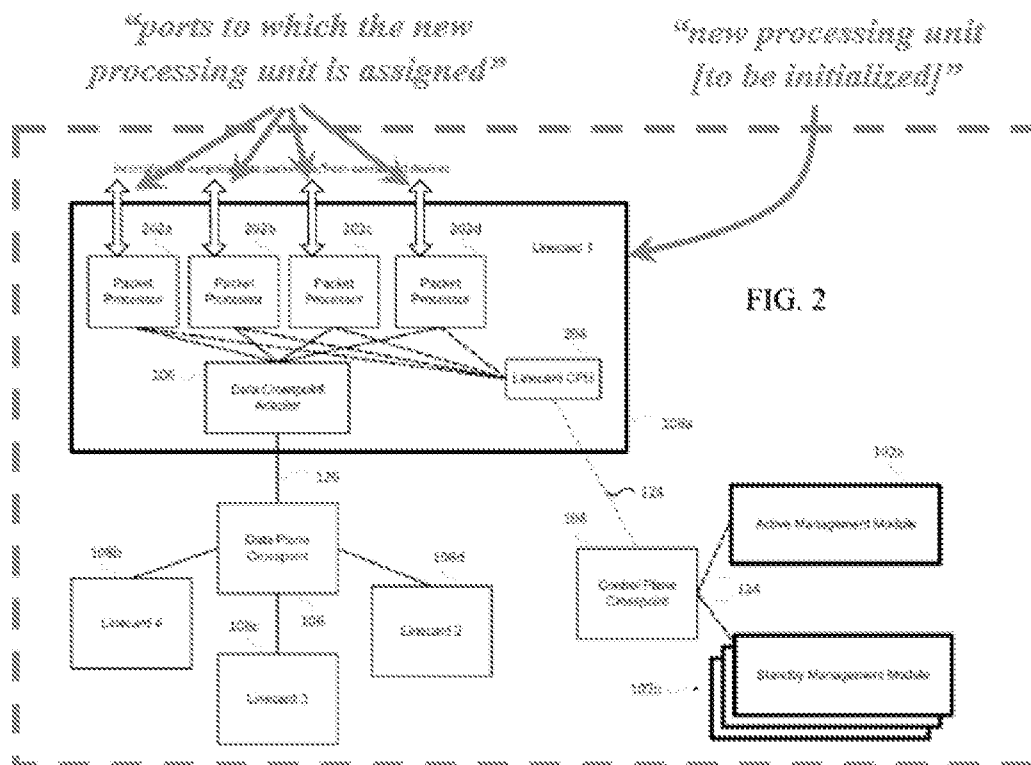
Accordingly, Naik discloses or suggests “**wherein operating the network element comprises initializing a new processing unit added to the network element,**” as recited in element [8.6]. *Id.* ¶ [0317].

xii. [8.6.1] “deactivating the ports to which the new processing unit is assigned;”

Naik in view of Ramanathan (Ex. 1010) would have rendered obvious the subject matter of element [8.6.1]. Ex. 1003 ¶ [0318].

Naik. Naik discloses **ports** (ports 256) to which a **new processing unit** (any of line cards 256 that “are individually hot-swappable”) may be assigned, but does not expressly disclose this **deactivating the ports**, as recited in element [8.6.1]. Ramanathan, however, discloses or suggests **deactivating** (“disabl[ing]” or “blocking” hardware access) **the ports** (physical port connector at arrows) **to which the new processing unit** (line card 1, e.g.) **is assigned**. *Id.*

Ramanathan. Ramanathan discloses a network switching device 100, indicating that its teachings are applicable to routers. Ex. 1010 ¶¶ [0033]-[0034]. Network switching device 100 includes a processor 102a and a plurality of line card modules 108a-108d interconnected by a control plane crosspoint 104, as in Naik. *Id.* at FIG. 1. Each of the line cards includes packet processors 202a-202d, each “configured with a physical port connector, schematically represented in FIG. 2 by the double arrows, for incoming and outgoing packets” *Id.* ¶ [0042].



As illustrated above in FIG. 2 (annotated outside dashed frame), an active processor 102a may be reset through a switchover to standby processor 102b. *Id.* ¶ [0064]. Similarly, one of the line cards, such as line card 1 above (i.e., “**the new processing unit**”), may perform a core reset and be rebooted, or initialized, after its line card processor (LP) 204 receives an upgrade message with an LP image. *Id.* ¶ [0080]. Upon rebooting with the new core, the LP—which corresponds to the claimed “**new processing unit**”—will initialize itself while blocking access to its hardware components, i.e., “**deactivating the ports to which the new processing unit is assigned,**” as recited in element [8.6.1]. *Id.* ¶ [0081] (“When the LP application is loaded, it will populate its software structures while **blocking the access to the HW components** during LP **initialization**, module configuration, and port configuration.”); Ex. 1003 ¶ [0320]. Those deactivated **ports to which the new processing unit is assigned** are circled in line card 1 in the annotated figure above.

Ramanathan explains the blocking of hardware access, or **deactivating the ports** of an LP, during initializing from an LP upgrade or reset in other contexts as well. *See id.* ¶ [0090] (“Before the LP upgrade begins, the hardware MAC table information is stored into a protected portion of memory. ... Until this process is completed, all hardware accesses may be disabled.”) ¶ [0092]

(“After an LP reset, the VLAN hardware accesses are disabled by the underlying system software. Once the LP application is up, hardware accesses are allowed.”).

For at least these reasons, Naik and Ramanathan discloses or suggest “**deactivating the ports to which the new processing unit is assigned,**” as recited in element [8.6.1]. Ex. 1003 ¶ [0322].

Motivation to Combine. A POSITA would have been motivated to apply the teachings of Ramanathan to disable or block hardware access to a processor being initialized when “hot-swap[ping]” a line card 256 of Naik into an active network with a reasonable expectation of success at least because doing so would help protect both the new line card and other components in the active network from erroneous messaging and data exchanges before the new line card is fully rebooted and ready for operation. *Id.* ¶ [0323].

The references provide teaching, suggestion, and motivation for their combination. MPEP § 2143.I; Ex. 1003 ¶ [0324]. For example, Naik discloses that its processing units—both route processors 210 and line cards 260—are “individually hot-swappable and may have redundant (standby backups).” Ex. 1010 ¶ [0045]. Those redundancies are illustrated, or at least suggested, in FIG. 2 of Naik, illustrating multiple instances of “STANDBY RP 210.” Therefore, Naik suggests to a POSITA that undisclosed steps will be needed to swap out or replace an active processing unit in edge router R1. Ex. 1003 ¶ [0324]. Thus, a POSITA would have been led to other references, like Ramanathan, addressing how to swap out, reboot, or otherwise initialize a new processing unit in an active network. *Id.*

Further, Ramanathan teaches detailed steps for swapping out, rebooting, or otherwise initializing processing units. One example in Ramanathan centers on a “switchover” between an active management processor (MP) 102a and a standby 102b “because it crashed, the processor locked up and a watchdog timer reset the processor, a user initiated the reset, etc.” Ex. 1010 ¶ [0064]. Ramanathan refers to this change as a “hot-switchover.” *Id.* ¶ [0066]. Another centers on a “core reset” of a line card module 108, which Ramanathan refers to as the line card being “hot-inserted.” *Id.* ¶¶ [0080]-[0082]. As a result, Ramanathan would have motivated a POSITA to implement, or at least explore, its teachings for use in initializing “hot-swappable” or “redundant (standby) backups” of processing units, such as in Naik. Ex. 1003 ¶ [0325].

In addition, combining Naik and Ramanathan also would have entailed the use of a known technique to improve a similar device in the same way. MPEP § 2143 at I(C); Ex. 1003 ¶ [0326].

Under that analysis, Naik discloses a “base” device as an edge router R1 having processing units that Naik states are “individually hot-swappable and may have redundant (standby backups).” Ex. 1005 ¶ [0045]; MPEP § 2143 at I(C)(1). Ramanathan presents a “comparable” device in terms of network switching device 100. Ex. 1010 at FIGS. 1, 2; MPEP § 2143.I(C)(2). Ramanathan discloses an “improvement” for the router, namely, in “techniques for facilitating software upgrade for a system such as a switching system or router” such as when undergoing a “hot-switchover” for a processing unit or the unit being “hot-inserted.” Ex. 1010 ¶ [0015]; MPEP § 2143.I(C)(3).

Moreover, a POSITA could have applied these “improvements” from Ramanathan into the edge router R1 of Naik with relatively minor software changes, such as within memory 230 of Naik, that would have led to predictable results for hot-swapping a new line card 260 within edge router R1 of Naik. Ex. 1003 ¶ [0327]; MPEP § 2143.I.(C)(3). For instance, a POSITA would have arrived at the combination of Naik and Ramanathan by modifying the software, such as within memory 230, of edge router R1 of Naik to deactivate ports of a new line card 260 being added or being rebooted within an active network. Ex. 1003 ¶ [0327]. In particular, the electronics associated with interface 270 in the new line card 260 could be controlled to block or deactivate communications on ports 256 associated with the new line card 260. This arrangement would have been a technically simple change for a POSITA to make without undue experimentation and with a reasonable expectation of success. *Id.*

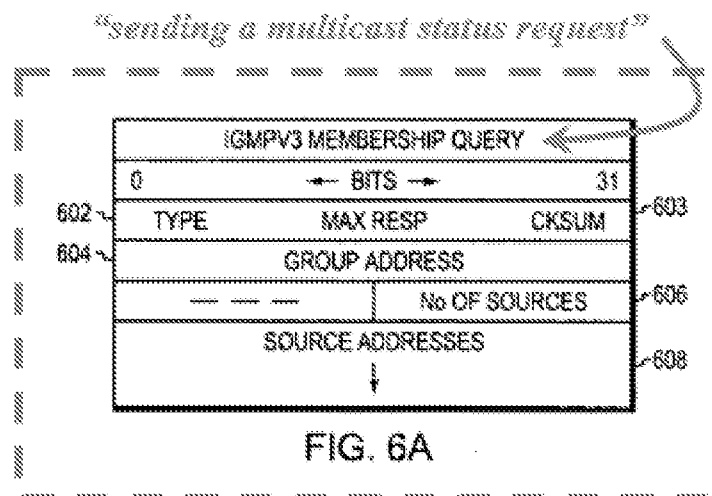
For at least these reasons, Naik and Ramanathan disclose or suggest all limitations of element [8.6.1], and Naik and Ramanathan would have rendered obvious “**deactivating the ports to which the new processing unit is assigned,**” as recited. Ex. 1003 ¶ [0328].

xiii. [8.6.2] “**while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request,**”

Naik in view of Ramanathan and Schulter discloses or suggests the subject matter of element [8.6.2]. *Id.* ¶ [0329].

Naik. Naik discloses or suggests **sending a multicast status request** (FIG. 6A), as claimed.

Naik explains that a Membership Query is a type of IGMPv3 message “where routers ask the multicast reception state of neighboring interfaces,” corresponding to **sending a multicast status request**, as recited in element [8.6.2]. Ex. 1005 ¶ [0060]; Ex. 1003 ¶ [0329].



According to Naik, FIG. 6A annotated above, is an IGMPv3 record packet format for a Membership Query to be sent by a router, such as edge router R1. The various records relate to different parameters of the query being sent. Ex. 1005 ¶ [0060].

While Naik discloses or suggests “**sending a multicast status request** (FIG. 6A),” it does not disclose doing so “**while the ports to which the new processing unit is assigned are deactivated**,” as recited in element [8.6.2]. Schulter (Ex. 1011) discloses, or at least suggests, that subject matter. Ex. 1003 ¶ [0330].

Schulter. Schulter discloses or suggests “**while the ports** (external network interface 129 in FIG. 1) **to which the new processing unit** (transmit control node being failed over with multicast MAC configuration) **is assigned are deactivated** (have no receive capability), **sending a multicast status request**,” as claimed. *Id.*

Schulter is directed to a virtual network system having a hardware platform with control nodes 120, as with “line cards 36” in the ’394 Patent. Each control node “is a single board that includes one or more (e.g., 4 processors, local memory, and local disk storage ...) ... that communicates with the Internet (or any other external network) 125 via an external network interface 129 having one or more Gigabit Ethernet NICs connected to Gigabit Ethernet links 121,123.” *Id.* at 3:34-44.

Schulter addresses different operating scenarios for two control nodes 120. In one scenario, both control nodes transmit packets to external nodes, but only one of the control nodes receives incoming packets, which enhances load balancing. Ex. 1011 at 20:1-3. According to Schulter, [t]he control node doing reception has IP information for filtering and multicast address information for

multicast MAC configuration.” *Id.* at 20:7-9. Because this information is needed to process incoming packets, should the receiving control node fail, the transmitting control node (“**the new processing unit**”) must be “failed over” and “configured with all MAC addresses on its physical interface to enable packet reception.” *Id.* at 20:18-22; Ex. 1003 ¶ [0332].

Until this MAC reconfiguring is complete, the ports of the transmitting control node remain incapable of receiving incoming packets. Ex. 1011 at 20:20-22 (all addresses must be configured “on its physical interface *to enable* packet reception”); Ex. 1003 ¶ [0333]. Additionally, Schulter explains that even if the control unit previously configured as transmit-only (the “**new processing unit**”) has the needed MAC addresses on its interfaces, its receiving ports are kept disabled until failover is complete: “Alternatively, both control nodes could have the same MAC address configured on their interfaces, but receives could be *physically disabled* on the Ethernet device by the device driver [“**the ports to which the new processing unit is assigned are deactivated**”] until an [sic] control node is ready to receive packets. *Then failover would simply enable receives on the device.*” Ex. 1011 at 20:22-27. These disclosures in Schulter to complete failover “**while the ports to which the new processing unit is assigned are deactivated**” are parallel to, and consistent with, the teachings of Ramanathan to reboot a processing unit “while blocking” or “disabl[ing]” hardware accesses until the unit is ready to operate. Ex. ¶¶ 1009 [0080]-[0081]; Ex. 1003 ¶ [0333].

Further, Schulter discloses “**sending a multicast status request**” while the ports are deactivated, as claimed. According to Schulter, virtual network drivers keep track of multicast group membership, such that “this information will always be available.” *Id.* at 20:32-35. Therefore, during failover, multicast tables can be rebuilt by querying these drivers: “Thus, a receive failover will result in multicast group membership being queried [i.e., “**sending a multicast status request**”] from virtual network drivers to rebuild the local multicast group membership tables.” *Id.* at 20:32-38. Being performed during the “receive *failover*”—when the receive control node has failed and the transmit control node (“**new processing unit**”) is being initialized with receiving capabilities—this process in Schulter takes place by definition “**while the ports to which the new processing unit is assigned are deactivated,**” as recited in element [8.6.2]. Ex. 1003 ¶ [0334].

For at least these reasons, Naik in view of Ramanathan and Schulter discloses or suggests **“while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request,”** as recited in element [8.6.2]. *Id.* ¶ [0335].

Motivation to Combine. A POSITA would have been motivated when installing and initializing a new line card for edge router R1 of Naik into an active network (often called a hot-swap, a switchover, or a failover) to send a multicast status request from other line cards of the edge router while the ports of the line card being initialized are deactivated as taught by Ramanathan and Schulter with a reasonable expectation of success at least because the forwarding table 400 within the new line card would need to be reconstructed with accurate information that could be readily obtained by sending a multicast status request to other devices. Ex. 1003 ¶ [0336].

The references provide teaching, suggestion, and motivation for their combination. *KSR*, 550 U.S. 398; MPEP § 2143.I; Ex. 1003 ¶ [0337]. For example, Naik discloses that its processing units—both route processors 210 and line cards 260—are “individually hot-swappable and may have redundant (standby backups).” Ex. 1010 ¶ [0045]. Those redundancies at least are illustrated in FIG. 2 of Naik, illustrating multiple instances of “STANDBY RP 210.” Therefore, Naik suggests to a POSITA that undisclosed steps will be needed to swap out or replace an active processing unit in edge router R1. Ex. 1003 ¶ [0337]. Thus, a POSITA would have been led to other references, like Schulter, addressing how to swap out, reboot, or otherwise initialize a new processing unit in an active network. *Id.*

Further, Schulter teaches detailed steps for swapping out, rebooting, or otherwise initializing processing units, which a POSITA would have recognized were applicable to Ramanathan’s deactivation process. *Id.* ¶ [0338]. For example, Schulter discloses “to ‘migrate’ to a new processor” or “to swap a new processor in to a PAN [processing area network] to replace a failing one.” Ex. 1011 at 28:5-12; *id.* at 28:31-32 (“When a new processor is ‘swapped in’ it will need to re-ARP to learn IP address to MAC address associations.”). Another example centers on “failover” or recovery of failed control nodes, i.e., interface or line card modules, within an active network. *Id.* at 20:1-46. At least these failover or recovery processes occur while the processor is still rebooting and not yet operating or active within the network. Ex. 1003 ¶ [0338]. As a result, Schulter would have motivated a POSITA to implement, or at least explore, its teachings for use in “swapping” in a new processor or performing a “failover” for a processing unit that has failed, such as one of line cards 260 in Naik. *Id.*

In addition, combining Naik and Schulter also would have entailed the use of a known technique to improve a similar device in the same way. MPEP § 2143.I(C); Ex. 1003 ¶ [0339]. Under that analysis, Naik discloses a “base” device as an edge router R1 having processing units that Naik states are “individually hot-swappable and may have redundant (standby backups).” Ex. 1005 ¶ [0045]; MPEP § 2143.I(C)(1). Schulter presents a “comparable” device in terms of hardware platform 100 with a series of control nodes 120. Ex. 1011 at FIG. 1; MPEP § 2143.I(C)(2). Schulter discloses an “improvement” for the router, namely, in querying “multicast group membership” when during failover and recovery of interface control nodes that are communicating multicast packet streams. Ex. 1011 at 20:1-46; MPEP § 2143.I(C)(3); Ex. 1003 ¶ [0339].

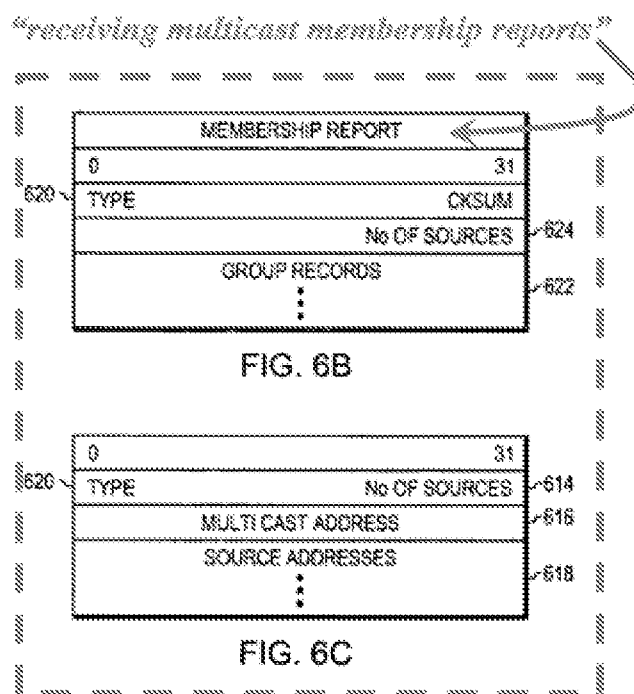
Moreover, a POSITA could have applied these “improvements” from Schulter into the edge router R1 of Naik with relatively minor software changes, such as within memory 230 of Naik, that would have led to predictable results for hot-swapping a new line card 260 within edge router R1 of Naik. Ex. 1003 ¶ [0340]; MPEP § 2143.I(C)(3). The change to implement this combination would have been technically insignificant—sending a membership query (as already taught by Naik) through other ports of edge router R1 of Naik during deactivation of ports—and been within the abilities of a POSITA to make without undue experimentation, leading to a reasonable expectation of success from the modification. Ex. 1003 ¶ [0340]. Indeed, Schulter itself suggests that its techniques are not difficult: “This operation[] is low overhead and requires no special processing except during failover and recovery, and doesn’t require any special replication of data between control nodes.” Ex. 1011 at 20:38-41.

For at least these reasons, Naik and Schulter disclose or suggest all limitations of element [8.6.2], and Naik, Ramanathan, and Schulter would have rendered obvious “**while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request,**” as recited. Ex. 1003 ¶ [0341].

xiv. **[8.6.3] “receiving multicast membership reports sent by multicast clients in response to the multicast status request,”**

Naik discloses or suggests the subject matter of element [8.6.3]. Namely, Naik discloses **receiving multicast membership reports** (Membership Reports in FIG. 6B) **sent by multicast clients** (IP systems) **in response to the multicast status request** (IGMPv3 Membership Query in FIG. 6A). Ex. 1003 ¶ [0342].

Naik explains that Membership Reports (i.e., “**multicast membership reports**”) “are sent by IP systems [i.e., “**multicast clients**”] to report the current multicast receptions state (or changes) and have the format of FIG. 6B,” following a Membership Query (“**multicast status request**”). Ex. 1005 ¶ [0061].



As shown above in annotated FIG. 6B, each group record 622 contains information “pertaining to the sender’s membership in a single multicast group on the interface from which the report is being sent.” *Id.* Each group record then has the format shown in FIG. 6C. *Id.*

Accordingly, Naik discloses or suggests “**receiving multicast membership reports sent by multicast clients in response to the multicast status request,**” as recited in element [8.6.3]. Ex. 1003 ¶ [0344].

- xv. [8.6.4] “**constructing the list by the new processing unit based on the received multicast membership reports; and**”

Naik in view of Schultzer would have rendered obvious the subject matter of element [8.6.4].

Naik. Naik discloses or suggests **the list** (MFIB 480 and MFIB Forwarding Table 400) **by the new processing unit** (a hot-swapped or backup line card 260) along with receiving **multicast membership reports** (Membership Reports in FIG. 6B). Ex. 1003 ¶ [0345]. Naik does not disclose

constructing the list ... based on the received multicast membership reports, as recited in [8.6.4].

Schulter. Schulter, however, discloses or suggests this subject matter, i.e., **constructing (rebuild) the list** (local multicast group membership tables) **by the new processing unit** (transmit control node being failed over with multicast MAC configuration) **based on the received multicast membership reports** (queried multicast group membership). Ex. 1003 ¶ [0346].

Schulter teaches the subject matter of element [8.6.4] in describing to rebuild group membership tables from membership reports during the failover of a receiving control node. According to Schulter, “a receive failover will result in multicast group membership being queried,” corresponding to “**sending a multicast status request**,” as claimed. Ex. 1011 at 20:35-38; Ex. 1003 ¶ [0347]. That query will be done “to rebuild the local multicast group membership tables,” which corresponds to “**constructing the list by the new processing unit based on the received multicast membership reports**,” as claimed. *Id.* And the querying and rebuild, or “**constructing**,” are done during deactivation of the ports of the new processing unit at least because it occurs when “a receiving control node fails” and during “a receive failover.” Ex. 1011 at 20:18-22, 20:28-38; Ex. 1003 ¶ [0347]. Therefore, Schulter discloses or suggests the subject matter of element [8.6.4].

Motivation to Combine. It would have been obvious to a POSITA considering Naik to have applied the teachings of Schulter regarding rebuilding or constructing its “local multicast group membership tables” to modify MFIB 480 and MFIB Forwarding Tables 400 based on received Membership Reports taught by Naik. Ex. 1003 ¶ [0348]. A new line card 260 that is “hot-swap[ped]” into an active network would need to have its forwarding tables built, and Schulter teaches one method for doing so that would have provided a reasonable expectation of success with minimal technical steps required within edge router R1 of Naik. *Id.*

As a result, Naik in view of Schulter would have rendered obvious “**constructing the list by the new processing unit based on the received multicast membership reports**,” in element [8.6.4]. *Id.* ¶ [0349].

xvi. [8.6.5] “after constructing the list, activating the ports to which the new processing unit is assigned.”

Naik in view of Ramanathan would have rendered obvious the subject matter of element [8.6.5]. Naik and Schulter do not expressly disclose **after constructing the list, activating the ports to which the new processing unit is assigned**. While Schulter teaches “constructing the

list” (element [8.6.4]), Ramanathan discloses or suggests after software update is complete, **activating the ports** (physical port connect at arrows) **to which the new processing unit** (line card 1, e.g.) **is assigned**, in element [8.6.5]. Ex. 1003 ¶ [0350].

As explained above for element [8.6.1], Ramanathan teaches **deactivating the ports to which the new processing unit is assigned** while a core reset is performed on a line card. *See* Ex. 1010 ¶¶ [0080]-[0081], [0090]-[0092]. According to Ramanathan, after the reset is complete and the line card has been initialized, its “hardware accesses are allowed,” which corresponds to the claimed “**activating the ports to which the new processing unit is assigned.**” *Id.* ¶ [0092] (“VLAN configuration is synchronized to the LP on card up event. After an LP reset, the VLAN hardware accesses are disabled by the underlying system software. *Once the LP application is up, hardware accesses are allowed.*”); Ex. 1003 ¶ [0351].

In one example for a reboot of a line card (i.e., an “LP core reset”), Ramanathan also discloses that its management processor (MP) “synchronizes the software MAC table” in the LP (i.e., “**after constructing the list**”) and keeps all hardware accesses “[u]ntil this process is completed” (i.e., thereafter “**activating the ports to which the new processing unit is assigned**”). *Id.* ¶ [0090] (“Before the LP upgrade begins, the hardware MAC table information is stored into a protected portion of memory. After the LP core reset, the MP synchronizes the software MAC table ... *Until this process is completed*, all hardware accesses may be disabled.”); Ex. 1003 ¶ [0352].

Therefore, Ramanathan discloses or suggests “**after constructing the list, activating the ports to which the new processing unit is assigned,**” as recited in element [8.6.5]. It would have been obvious to a POSITA considering Naik to have applied the teachings of Ramanathan regarding activating the ports to which the new processing unit is assigned after constructing the list. Ex. 1003 ¶ [0353]. Activating deactivated ports after software is rebuilt, i.e., “after constructing the list,” would have been a logical and necessary step to place the new processing unit in an operating condition. *Id.* Indeed, a POSITA would have understood that a hot-swap of a line card in Naik, and the necessary update and initialization of the processing unit and its software, would not have been complete and ready for operation until after forwarding tables 400 were rebuilt or “constructed.” *Id.* Therefore, a POSITA would have been motivated to wait unit “**after constructing the list**” to “**activat[e] the ports to which the new processing unit is assigned**” and have had a reasonable expectation of success in doing so. *Id.*

In summary, Naik in view of Reader, further in view of Ramanathan and Schulter, would have rendered independent claim 8 unpatentable under 35 U.S.C. § 103(a) (pre-AIA). Ex. 1003 ¶ [0354].

b. Dependent Claim 9

- i. **[9.1] “The method according to claim 8, wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over a given port only when a number of the multicast packet streams that are permitted for forwarding over the given port, including the given multicast packet stream, does not exceed a maximum value specified for the given port.”**

Naik discloses or suggests the subject matter of claim 9. Specifically, Naik discloses **wherein updating the list** (MFIB Forwarding Tables 400 within line cards 260) **comprises modifying the list** (MFIB Forwarding Tables 400 are “populated” from MRIB) **to indicate that the given multicast packet stream** (collective entries 410 in MFIB Forwarding List 400 in FIG. 4) **is permitted for forwarding over a given port only when the number of the multicast packet streams that are permitted for forwarding** (tracking table 514, number of channels joined from each port, [0082]) **over the given port, including the given multicast packet stream, does not exceed the upper limit** (maximum number of “joins” allowed by the switch for “LIMIT” action; deny admission 510) **specified for the given port**. Ex. 1003 ¶ [0355].

For “**modifying the list to indicate the given multicast packet stream is permitted for forwarding**,” Naik teaches modifying its MFIB forwarding tables within line cards 260 based on routing maps. Ex. 1005 ¶ [0045]; Ex. 1003 ¶ [0356]. As noted above for element [8.4.1], Naik teaches that its forwarding tables 400 within MFIB 480, shown in FIG. 2 within each line card 260, are “populated” with routing information from MRIB 238 within memory 230 about how to forward data packets. *Id.* (the MFIB derived from the MRIB “is embodied as one or more multicast forwarding tables 400 that are populated on the line cards 260 and whose contents describe how to forward data packets through the router.”). A POSITA would understand that this populating of the forwarding tables 400 indicates that “**the given multicast packet stream is permitted for forwarding**” to ensure the line cards can properly forward the requested multicast packet stream as it arrives. Ex. 1003 ¶ [0356]. Indeed, Naik teaches: (“In response to receiving the incoming packet from a neighbor, the MFIB 480 performs a lookup into its forwarding table 400 to find a

route of an entry 410 that matches a destination address of the packet. The matching route instructs the router as to which interfaces the packet should be forwarded.”). Ex. 1005 ¶ [0052]

Conversely, Naik discloses that the forwarding table 400 may also be modified—by removing a requestor from a multicast group—to indicate that a multicast packet stream is *not* permitted for forwarding to that requestor. For instance, if ACM 500 concludes after applying a traffic map and a policy map that a multicast request should be denied, Naik discloses that “if already a member[,] the requestor is removed from the group *and/or from the forwarding table.*” *Id.* ¶ [0074]. Therefore, a POSITA would understand from Naik that the presence of a multicast packet stream within forwarding table 400, and the modification of forwarding table 400, indicates that a **given multicast packet stream is permitted for forwarding**. Ex. 1003 ¶ [0357].

For forwarding “**only when the number of the multicast packet streams that are permitted for forwarding over the high-capacity port, including the given multicast packet stream, does not exceed the upper limit,**” Naik teaches that ACM 500 accepts a multicast request—and therefore authorizes modification of the MFIB—only if the given multicast packet stream of the request would not exceed the upper limit, or maximum value, set for a port. *Id.* ¶ [0358]. For instance, ACM 500 tracks the number of streams for a port in a tracking table, compares the number to a limit, and denies any further multicast join requests if the limit would be exceeded:

The ACM would track the number involved and compare it to the limit. When the limit is reached any additional joins will always be denied regardless of the traffic and service policy. The limit parameter may be stored in the tracking table described below, but in any event the PIM 234 and the MRIB 238 may participate in implementing such limits.”

Ex. 1005 ¶¶ [0072], [0055]. Thus, requests are accepted “**only when the number ... does not exceed the upper limit,**” as claimed. Naik at Abstract (“Incoming packet traffic received by an edge router is snooped and, when a request to join a multicast channel is received, the traffic is analyzed. ... Also, the actions may be to limit the number of routing states, by denying admittance to a groups [sic] once a limit number of requests to join, or other such parameter, is reached.”); Ex. 1003 ¶ [0358].

Consequently, Naik discloses or suggests “**wherein updating the list comprises modifying the list to indicate that the given multicast packet stream is permitted for forwarding over a given port only when a number of the multicast packet streams that are**

permitted for forwarding over the given port, including the given multicast packet stream, does not exceed a maximum value specified for the given port,” as claimed. Ex. 1003 ¶ [0359]; Ex. 1002 at 112.

c. Independent Claim 18

i. [18.pre] “A network element, comprising;”

Naik discloses or suggests the subject matter of element [18.pre]. *See supra* Discussion of Naik in Ground 1, [1.pre], which is incorporated by reference.

ii. [18.1] “multiple ports; and”

Naik discloses or suggests the subject matter of element [18.1]. *See supra* Discussion of Naik in Ground 1, [1.1], which is incorporated by reference.

iii. [18.2] “multiple interconnected processing units, [18.2.1] wherein each processing unit is assigned to a different subset of the multiple ports of the network element, and”

Naik discloses the subject matter of elements [18.2], [18.2.1]. *See supra* Discussion of Naik in Ground 1, [1.1], [1.1.1], and [1.1.2], which is incorporated by reference.

iv. [18.2.2] “wherein each processing unit contains a processor for processing packets communicated over the different subset of the multiple ports to which the processing unit is assigned,”

Naik discloses or suggests the subject matter of element [18.2.2]. *See supra* Discussion of Naik in Ground 1, [1.1.3], which is incorporated by reference.

v. [18.2.3] “wherein the processing units are configured to process packets that are communicated over respective subsets of one or more of the ports,”

Naik discloses or suggests the subject matter of element [18.2.3]. *See supra* Discussion of Naik in Ground 1, [1.1.3], which is incorporated by reference.

vi. [18.3] “to store respective lists,” [18.3.1] “each of which indicates all of one or more multicast packet streams that are permitted for forwarding by the network element and”

Naik discloses or suggests the subject matter of elements [18.3], [18.3.1]. *See supra* Discussion of Naik in Ground 1, [1.2] and [1.2.1], which is incorporated by reference.

- vii. **[18.3.2] “further indicates which of the one or more ports multicast packet stream is permitted to be forwarded to,”**

Naik discloses or suggests the subject matter of element [18.3.2]. *See supra* Discussion of Naik in Ground 1, [1.2.2], which is incorporated by reference.

- viii. **[18.4] “to receive from a client a request relating to reception of a given multicast packet stream, and”**

Naik discloses or suggests the subject matter of element [18.4]. *See supra* Discussion of Naik in Ground 1, [1.3], which is incorporated by reference.

- ix. **[18.5] [18.5.1] “to distribute the request among the processing units within the network element”**

Naik discloses or suggests the subject matter of elements [18.5], [18.5.1]. *See supra* Discussion of Naik in Ground 1, [1.4] and [1.4.1], which is incorporated by reference.

- x. **[18.5.2] so as to update the lists responsively to the distributed request and”**

Naik, or Naik in view of Reader, discloses the subject matter of element [18.5.2]. *See supra* Discussion of Naik and Reader in Ground 1, [1.4.2], which is incorporated by reference.

- xi. **[18.5.3] “to selectively forward multicast packets associated with the given multicast packet stream in accordance with the updated lists,”**

Naik, or Naik in view of Reader, discloses the subject matter of element [11.5.2]. *See supra* Discussion of Naik and Reader in Ground 1, [1.5], which is incorporated by reference.

- xii. **[18.6] “wherein a new processing unit added to the network element is configured to initialize by:”**

Naik discloses or suggests the subject matter of element [18.6]. *See supra* Discussion of Naik in Ground 3, [8.6], which is incorporated by reference.

- xiii. **[18.6.1] “deactivating the ports to which the new processing unit is assigned;”**

Naik in view of Ramanathan discloses or suggests the subject matter of element [18.6.1]. *See supra* Discussion of Naik and Ramanathan in Ground 3, [8.6.1], which is incorporated by reference.

- xiv. **[18.6.2] “while the ports to which the new processing unit is assigned are deactivated, sending a multicast status request,”**

Naik in view of Ramanathan and Schulter disclose or suggest the subject matter of element [18.6.2]. *See supra* Discussion of Naik, Ramanathan, and Schulter in Ground 3, [8.6.2], which is incorporated by reference.

- xv. **[18.6.3] “receiving multicast membership reports sent by multicast clients in response to the multicast status request,”**

Naik discloses or suggests the subject matter of element [18.6.3]. *See supra* Discussion of Naik in Ground 3, [8.6.3], which is incorporated by reference.

- xvi. **[18.6.4] “constructing the list by the new processing unit based on the received multicast membership reports; and”**

Naik and Schulter disclose or suggest the subject matter of element [18.6.4]. *See supra* Discussion of Naik and Schulter in Ground 3, [8.6.4], which is incorporated by reference.

- xvii. **[18.6.5] “after constructing the list, activating the ports to which the new processing unit is assigned.”**

Naik and Ramanathan disclose or suggest the subject matter of element [18.6.5]. *See supra* Discussion of Naik and Ramanathan in Ground 3, [8.6.5], which is incorporated by reference.

2. Ground 4: Naik in View of Reader, Ramanathan, and Schulter, Further in View of Deng Would Have Rendered Obvious Dependent Claim 10

- a. **10. [10.1] “The method according to claim 8, wherein updating the list comprises tracking a use of a permission to forward a certain multicast packet stream over a certain port by monitoring requests relating to the certain multicast packet stream that are received at the network element, and”**

Naik discloses or suggests the subject matter of element [10.1]. *See supra* Discussion of Naik in Ground 2, [16.1], which is incorporated by reference.

- b. **[10.2] “removing the permission from the list responsively to determining that the permission is aged.”**

Deng discloses or suggests the subject matter of element [10.2]. *See supra* Discussion of Deng in Ground 2, [16.2], which is incorporated by reference.

C. Secondary Considerations

This Request demonstrates that the Challenged Claims of the '394 Patent are unpatentable as having been obvious in view of the prior art references. The Applicant did not identify any evidence of secondary considerations during prosecution. Further, the clear teachings in the prior outweigh any supposed "secondary considerations." Ex. 1003 ¶ [0379]; *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 36 (1966).

V. Disclosure of Concurrent Litigation, Reexamination, and Related Proceedings

Based on information available to Requester, the '394 Patent is the subject of District Court litigations, as listed in the below table. Requester is unaware of any prior reexaminations or other post-grant proceedings in which the '394 Patent is or has been involved.

- *Portsmouth Network Corporation v. Juniper Networks, Inc.*, Case No. 1-23-cv-12201, USDC, DMA, filed September 26, 2023
- *Portsmouth Network Corporation v. Cisco Systems, Inc.*, Case No. 2:23-cv-00441, USDC, EDTX, filed September 26, 2023

VI. Conclusion

The Commissioner is hereby authorized to charge Deposit Account 12-0769 under Docket No. U022-0012RE the *Ex Parte* Reexamination fee of \$12,600 under 37 C.F.R. § 1.20(c)(2). Requester believes no other fee is due with this submission, however the Commissioner is hereby authorized to charge any fee deficiency or credit any over-payment to Deposit Account 12-0769.

Please direct all correspondence in this matter to the undersigned.

Respectfully submitted,

Dated: March 5, 2024

By: /R. Bruce Bower/
R. Bruce Bower

James D. Stein (Reg. No. 63,782)
R. Bruce Bower (Reg. No. 37,099)
Lee & Hayes, P.C.
75 14th Street N.E., Suite 2500
Atlanta, Georgia 30309
bruce.bower@leehayes.com
P: 404-736-1924

Counsel for Requester Unified Patents, LLC

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

<i>In re</i> patent of Rafi Ram <i>et al.</i>	§ Attorney Docket No.: U022-0012RE
	§
U.S. Patent 8,014,394B2	§
	§
Issue Date: September 6, 2011	§ Customer No.: 29150
	§
Filing Date: February 11, 2009	§
	§
For: HIGH-SPEED PROCESSING OF	§
MULTICAST CONTENT REQUESTS	§

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 1.248 and 1.550 and MPEP § 2266.03, the undersigned attorney for the Requester certifies that a copy of the foregoing **REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT 8,014,394** and accompanying Exhibits Ex. 1001 to Ex. 1011 were served by First Class U.S. Mail or equivalent at the following address of record for the subject patent (at left) and as a courtesy to litigation counsel for the subject patent (at right):

Patent Owner
May Patents Ltd.
c/o Dorit Shem-Tov
P.O.B 7230
Ramat-Gan 5217102
ISRAEL

Litigation Counsel
James R. Nuttall
Katherine H. Tellez
Robert F. Kappers
Daniel F. Gelwicks
STEPTOE & JOHNSON LLP
227 West Monroe Street, Suite 4700
Chicago, IL 60606

Respectfully submitted,

Dated: March 5, 2024

By: /R. Bruce Bower/
R. Bruce Bower (Reg. No. 37,099)
Lee & Hayes, P.C.
75 14th Street N.E., Suite 2500
Atlanta, Georgia 30309
bruce.bower@leehayes.com
P: 404-736-1924