IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Bechtolsheim & Cheriton
U.S. Patent No.: 6,377,577
Issue Date: Apr. 23, 2002
Appl. Serial No.: 09/108,071
Filing Date: June 30, 1998
Title: ACCESS CONTROL LIST PROCESSING IN HARDWARE

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

PETITION FOR INTER PARTES REVIEW OF UNITED STATES PATENT NO. 6,377,577 PURSUANT TO 35 U.S.C. §§ 311–319, 37 C.F.R. § 42
TABLE OF CONTENTS

I. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1) .......................1
   A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1) ....................1
   B. Related Matters Under 37 C.F.R. § 42.8(b)(2) ..........................1
   C. Lead and Back-Up Counsel and Service Information ...............1

II. PAYMENT OF FEES — 37 C.F.R. § 42.103 .................................2

III. REQUIREMENTS FOR IPR UNDER 37 C.F.R. § 42.104 ...............2
   A. Grounds for Standing Under § 42.104(a) ...............................2
   B. Challenge Under § 42.104(b) and Relief Requested .................2

IV. SUMMARY OF THE ’577 PATENT .............................................3
   A. Brief Description .................................................................3
   B. Level of Ordinary Skill in the Art .........................................4

V. CLAIM CONSTRUCTION .............................................................4
   1. “access control” .................................................................6
   2. “associative memory” .........................................................7
   3. “packet label” .................................................................8
   4. “responsive” .................................................................9
   5. “megapacket” .................................................................9

VI. MANNER OF APPLYING CITED PRIOR ART TO EVERY CLAIM FOR WHICH IPR IS REQUESTED, THUS ESTABLISHING A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE ’577 PATENT IS UNPATENTABLE .................................10
A. [Ground 1] — Claims 1, 2, 5, 7–10, 12–16, 28, and 29 are anticipated by Feldmeier under 35 U.S.C. § 102 .......................................................... 14


C. [Ground 3] — Claims 22, 28, 29, and 31 are obvious over Feldmeier in view of Muller under 35 U.S.C. § 103 ............................................. 38

D. [Ground 4] — Claim 30 is obvious over Feldmeier in view of Muller and the knowledge of one of ordinary skill in the art under 35 U.S.C. § 103 ................................................................................. 43


VII. THIS PETITION DOES NOT PRESENT REDUNDANT GROUNDS FOR INSTITUTION ........................................................................ 45

VIII. CONCLUSION ...................................................................................... 47
**EXHIBITS**

<table>
<thead>
<tr>
<th>Exhibit Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARISTA-1002</td>
<td>Excerpts from ’577 Patent File History</td>
</tr>
<tr>
<td>ARISTA-1003</td>
<td>Declaration of H. Jonathan Chao, Ph.D.</td>
</tr>
<tr>
<td>ARISTA-1004</td>
<td>Curriculum Vitae of H. Jonathan Chao, Ph.D.</td>
</tr>
<tr>
<td>ARISTA-1005</td>
<td>U.S. Patent No. 5,920,886 (“Feldmeier”)</td>
</tr>
<tr>
<td>ARISTA-1006</td>
<td>A. McAuley &amp; P. Francis, <em>Fast Routing Table Lookup Using CAMs</em> (IEEE 1993) (Incorporated by Feldmeier)</td>
</tr>
<tr>
<td>ARISTA-1007</td>
<td>U.S. Patent No. 6,081,522 (“Hendel”)</td>
</tr>
<tr>
<td>ARISTA-1008</td>
<td>U.S. Patent No. 5,938,736 (“Muller”)</td>
</tr>
<tr>
<td>ARISTA-1009</td>
<td>U.S. Patent No. 3,602,899 (“Lindquist”)</td>
</tr>
<tr>
<td>ARISTA-1010</td>
<td>Tong-Bi Pei et al., <em>VLSI Implementation of Routing Tables: Tries and CAMs</em> (IEEE 1991) (“Pei”)</td>
</tr>
<tr>
<td>ARISTA-1013</td>
<td>L. Chisvin et al., <em>Content-Addressable and Associative Memory: Alternatives to the Ubiquitous RAM</em> (IEEE 1989) (“Chisvin”)</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARISTA-1016</td>
<td>M. Ranum, <em>A Network Firewall</em> (Digital Equipment Corporation June 12, 1992)</td>
</tr>
<tr>
<td>ARISTA-1017</td>
<td>U.S. Patent No. 5,841,874 (“Kempke”)</td>
</tr>
<tr>
<td>ARISTA-1019</td>
<td>U.S. Patent No. 5,867,495 (“Elliott”)</td>
</tr>
<tr>
<td>ARISTA-1020</td>
<td>Excerpts from Radia Perlman, <em>Interconnections: Bridges and Routers</em> (1992)</td>
</tr>
<tr>
<td>ARISTA-1021</td>
<td>[ RESERVED ]</td>
</tr>
<tr>
<td>ARISTA-1022</td>
<td>[ RESERVED ]</td>
</tr>
<tr>
<td>ARISTA-1023</td>
<td>Revised Claim Construction Proposals</td>
</tr>
</tbody>
</table>

I. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

Arista is the real party-in-interest.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)


C. Lead and Back-Up Counsel and Service Information

Petitioner designates W. Karl Renner, Reg. No. 41,265, as Lead Counsel, and Kevin E. Greene, Reg. No. 46,031, as Backup Counsel, both available at 3200
RBC Plaza, 60 South Sixth Street, Minneapolis, MN 55402 (T: 202-783-5070; F: 202-783-2331), or electronically by email at IPR40963-0003IP2@fr.com.

II. PAYMENT OF FEES — 37 C.F.R. § 42.103

Petitioner authorizes the Patent and Trademark Office to charge Deposit Account No. 06-1050 for the fee set in 37 C.F.R. § 42.15(a) for this Petition and further authorizes for any additional fees to be charged to this Deposit Account.

III. REQUIREMENTS FOR IPR UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under § 42.104(a)

Petitioner certifies that the ’577 Patent is available for IPR. The petition is being filed within one year of service on Petitioner of a complaint alleging infringement of the Patent. Petitioner is not barred or estopped from requesting this review on the below-identified grounds.¹

B. Challenge Under § 42.104(b) and Relief Requested

Petitioner requests IPR of the Challenged Claims on the grounds set forth in the table below, and requests that the Challenged Claims be found unpatentable.

An explanation of unpatentability is provided, indicating where each element is found in the prior art. Additional explanation and support for each ground is set forth in Ex. 1003, Declaration of H. Jonathan Chao, Ph.D.

<table>
<thead>
<tr>
<th>Ground</th>
<th>'577 Patent Claims</th>
<th>Basis for Rejection</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 2, 5, 7–10, 12–16, 28, 29</td>
<td>Feldmeier — 35 U.S.C. § 102</td>
</tr>
<tr>
<td>3</td>
<td>22, 28, 29, 31</td>
<td>Feldmeier, Muller — 35 U.S.C. § 103</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>Feldmeier, Muller, knowledge of one of ordinary skill — 35 U.S.C. § 103</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>Feldmeier, Muller, Elliott — 35 U.S.C. § 103</td>
</tr>
</tbody>
</table>

The ’577 Patent issued from U.S. application number 09/108,071, which was filed on June 30, 1998. The ’577 Patent does not include a priority claim.

Feldmeier (Ex. 1005), Hendel (Ex. 1007), Muller (Ex. 1008) and Elliott (Ex. 1019) are prior art under 35 U.S.C. § 102(e). Hendel was filed on June 30, 1997, Feldmeier was filed on March 14, 1997, Muller was filed on June 30, 1997, and Elliott was filed on November 18, 1996.

IV. SUMMARY OF THE ’577 PATENT

A. Brief Description

The ’577 Patent describes techniques for enforcing access control for a computer network using a content-addressable memory (CAM). Ex. 1001, at (57). Information from an incoming packet is matched against certain access control
information stored in the CAM. *Id.* The CAM searches all stored information at once. The access control information includes a priority value, either inherently based on location in the CAM, or as a separate, discrete value. When a search is performed, matches are analyzed according to priority, and an access control decision is made. This access control decision is used to make a decision for routing. *Id.; see also id. 7:34–46 (claim 1); see generally Ex. 1003, ¶¶ 39–41.*

**B. **Level of Ordinary Skill in the Art

A person of ordinary skill in the art as of the Filing Date would have had at least a master’s degree (or a substantively equivalent degree) in electrical engineering, computer engineering, or computer science (or a substantively related field), in addition to two years of post-graduate work experience, whether in industry or conducting research, in the field of networking. Additional education in a relevant field, such as computer science, computer engineering, or electrical engineering, or industry experience may compensate for a deficit in one of the other aspects of the requirements stated above. Ex. 1003, ¶¶ 22–24.

**V. **CLAIM CONSTRUCTION

A claim in IPR is given its “broadest reasonable construction in light of the specification of the patent in which it appears.” 37 C.F.R. § 42.100(b). For this
proceeding only, Petitioner submits constructions for the following terms.\(^2\) All remaining terms should be given their broadest reasonable plain meaning.\(^3\)

\[^2\] In several instances, the constructions set forth herein derive from positions taken by Cisco in its efforts to assert the ’577 Patent. Petitioner does not necessarily accept that such constructions are either reasonable or correct in view of the ’577 Patent, its file history, and the relevant extrinsic evidence. Petitioner expressly reserves the right to demonstrate, in appropriate forums outside of this proceeding, that Cisco’s claim interpretations are neither reasonable nor correct.

\[^3\] Petitioner’s proposals are for the sole purpose of determining whether the prior art anticipates or renders obvious the Challenged Claims. Petitioner does not concede that any claim meets statutory standards for validity. Petitioner reserves all rights to contend that one or more Challenged Claims are invalid for reasons out of scope for IPR, including but not limited to lack of definiteness under §112, ¶2 and lack of written description under §112, ¶1. Definiteness and description problems in the claims are no bar to IPR, and may be set aside to permit review under §§102 and 103. \textit{E.g.}, \textit{Vibrant Media, Inc. v. Gen’l Elec. Co.}, IPR2013-00172, 2014 WL 3749773, at *6–7 (Patent Tr. & App. Bd. July 28, 2014).
1. **“access control”**

“Access control” appears throughout the specification and claims of the ’577 Patent. A person of skill in the art would have understood “access control” to encompass restrictions or modifications of the transmission of a packet. According to the patent, the restriction of the transmission of messages is one form of “access control.” Ex. 1001, 1:4–10. The patent also identifies as “access control” modifications of the routing decisions: “The invention [which includes access control, as it appears in each and every claim] can be used to augment or override routing decisions otherwise made by the router, so as to implement QOS (quality of service), and other administrative policies, using the [CAM].” Ex. 1001, 6:10–19 (emphasis added). A person of skill would appreciate that “access control” in the field has broad applicability concerning restriction or modification of packet transmission and its use in the ’577 Patent thus includes concepts such as filtering, firewalls, security, identification of allowed ports, quality of service, network administrative policies, denying access, and many other similar actions. *See also* Ex. 1003, ¶¶ 43–44. Accordingly, one of ordinary skill at the time of the ’577 Patent would have understood “access control” to be broad enough to include “restrictions or modifications of the transmission of a packet.”
2. “associative memory”

The ’577 Patent claims recite “associative memory.” An associative memory is a memory that stores and searches on values, like a database, rather than on memory locations as in traditional memory. Ex. 1003, ¶ 45. The specification teaches a specific type of associative memory implemented in hardware known as “CAM” (content-addressable memory), configured such that a search of the CAM’s entries will search every entry at once. Id. In the “Summary of the Invention,” the ’577 Patent states:

A sequence of access control specifiers from an ACL are recorded in a CAM, and information from the packet header is used to attempt to match selected source and destination IP addresses or subnets, ports, and protocols, against all the ACL specifiers at once. Ex. 1001, 2:40–44 (emphasis added); see also id. 4:34–35 (defining “CAM” as “content-addressable memory”). This same reference is seen in the patent abstract (id. at (57)), and throughout the figures and specification (id. 2:51–66 (describing preferred embodiments, each using a CAM), 4:34–36 (describing how figure 2 depicts a CAM). Dependent claims recite different implementations of an “associative memory,” including CAMs. Id. 7:59–62 (in claim 5, addressing a “hardware content-associative memory”), 7:63–8:3 (same, for claim 6), 8:4–5 (in claim 7, referring to a “ternary content-associative memory”).
Further, a person of skill in the art would have understood that the patent phrase “associative memory” includes a content-addressable memory. Ex. 1003, ¶¶ 45–46. For purposes of this proceeding, the term “associative memory” should be construed broadly enough to include a “content-addressable memory (CAM).”

3. “packet label”

The ’577 Patent describes the “packet label” as a “collection” of “a set of selected elements of a packet header 133 for each packet.” Ex. 1001, 4:2–3. Alternatively, the patent otherwise describes a packet label as “any collection of information derived from the packet (preferably from the packet header) used for access control.” Id. 4:10–12. The patent is replete with references to the “packet header,” and these references are consistent. Id. at (57), 2:40–44 (describing, in the Summary of the Invention, use of the “packet header”), fig.2 (depicting “packet header 133”), fig.3 (depicting step 322, “identify header,” and 323, “select label”), and accompanying text. In light of this disclosure, a person of ordinary skill in the art would conclude that the broadest reasonable interpretation of “packet label” includes “information related to the packet, such as the packet header.” Ex. 1003, ¶¶ 47–48. Arista thus submits that, for purposes of this proceeding, the term “packet label” should be construed broadly enough to include “information related to the packet, such as the packet header.”
4. “responsive”

It is appropriate for the PTAB to interpret this term consistent with the interpretation implicit in Cisco’s assertion of the ’577 Patent. In pending litigation before the International Trade Commission (ITC), Cisco has proposed that “responsive” be construed to mean “derived from.” See Ex. 1023, at 3 (setting forth Cisco’s proposed construction of, inter alia, “responsive”); see also Ex. 1003, ¶ 49. Application of the broadest reasonable interpretation standard should account for patentee Cisco’s proposed definition. Arista thus submits that, for purposes of this proceeding only, “responsive” should be construed broadly enough to include “derived from.”

5. “megapacket”

The term “megapacket” does not appear in the abstract, figures, or written description of the ’577 Patent. A person of skill in the art would have appreciated that the prefix “mega-” generally imposes, in computer science circles, a multiplier of $10^6$ (i.e., 1 million) to the modified unit, “packet.” Ex. 1003, ¶¶ 53–54. Such a person would conclude that the term “megapacket” means “one million packets.” Arista thus submits that, for purposes of this proceeding, the term “megapacket” should be construed broadly enough to include “one million packets.”
VI. MANNER OF APPLYING CITED PRIOR ART TO EVERY CLAIM FOR WHICH IPR IS REQUESTED, THUS ESTABLISHING A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE ’577 PATENT IS UNPATENTABLE

The ’577 Patent is directed to the straightforward concept of using previously-known memory technologies called content addressable memories (“CAMs”) in the already-mature field of network access control. See Ex. 1001, at (57). The ’577 Patent does not claim to have invented either of these technologies, nor did Mr. Bechtolsheim or Dr. Cheriton (the named inventors) actually invent CAMs or the field of network access control. As the ’577 Patent’s written description itself acknowledges, access control technology was a well-developed field at the time. Id. 1:4–2:2 (describing previously-available technologies for access control, including the use of “ACLs (access control lists)”). The use of CAMs, and even TCAMs, with access control functionality was known in the industry almost two years prior to the filing date. Ex. 1017 (U.S. Patent No. 5,841,874 (filed Aug. 13, 1996), 1:58–63 (disclosing use of “ternary CAMs . . . for such things as address resolution [and] filtering”); Ex. 1003, ¶ 32–38 (discussing same). Indeed, the ’577 Patent’s sole basis for alleging patentability was an apparent understanding—albeit an incorrect one—that using CAMs for network access control was neither previously known in the field nor obvious to those of ordinary skill. Petitioner submits that the enclosed references readily reveal that others had previously developed such technology. As such, none of the
Challenged Claims was either new or nonobvious in the field as of the '577 Patent’s filing date.

**Feldmeier** describes techniques for a networking device to store certain information useful for routing and for access control in a CAM. Ex. 1005, 13:19–21, 5:27–29; see also Ex. 1003, ¶¶ 56–57 (describing Feldmeier in detail). Specifically, Feldmeier describes storing information about network structure, and about which communication should be permitted on which data paths. Feldmeier specifically describes using a CAM to aid in hierarchical address translation (i.e., routing packets to the correct network based on the hierarchical organization of network addresses), but discloses alternate uses for its technology in the field of access control. When a packet is received, Feldmeier teaches reviewing the packet header information and performing a lookup against the information stored in the CAM. Ex. 1005, fig.14 & 13:24–27. Based on the information in the CAM, a decision is made about whether to pass the packet forward, or discard it. *Id.* Feldmeier also teaches choosing between multiple matches using the relative priority of the various matched entries in the CAM. *Id.* 11:10–12.

**Hendel** describes a network routing and access control system in which forwarding logic in a network device compares multiple aspects of an incoming packet’s header data simultaneously to data stored in a CAM so as to make appropriate determinations for how and whether to forward an incoming packet.
See generally Ex. 1003, ¶¶ 88–90 (discussing Hendel in detail).

Specifically, Hendel teaches using “forwarding logic” to compare aspects of packet information to different layer 2 and 3 data stored in a CAM, and then using “merge logic” to merge the results of those comparisons into a final routing decision. Hendel describes its approach using the OSI layer model. Ex. 1007, 2:5–39. It performs packet header analysis in both “L2 Logic 62” (concerning link layer packet information) and “L3 Logic 64” (concerning network layer packet information). Id. 10:12–33. These compare L2 and L3 information for the packet, respectively, to information stored in forwarding memory 40, which is a CAM. Id. 8:57–61. Additional packet analysis occurs in Class Logic 60. Id. 11:49–60. Merge logic 66 receives information relating to both layers, as well as information from class logic 60, and determines whether and how to route the packet. Id. 10:47–52.

For example, Hendel discusses using its techniques for network firewall applications. Id. 14:8–12. It teaches that merge logic 66 may, in some circumstances, instruct that packets be discarded where their proposed destination
is inappropriate. *Id.* 10:51–52. Hendel also teaches techniques for promoting prioritization decisions among the various entries stored in the CAM. *Id.* 8:37–42.

**Muller** describes a network switch capable of examining incoming packets and comparing them to a search database, particularly a CAM. Ex. 1008, 2:40–64; *see also* Ex. 1003, ¶¶ 100–02 (discussing Muller in detail).

Like the network device described in Hendel, Muller’s device examines packets at multiple layers of the OSI layer model, thus enabling a number of features including, but not limited to, filtering packets for access control purposes. Ex. 1008, 5:2–5. Muller teaches separate process flows for analyzing a packet’s layer 2 (i.e., data link layer) header information, and analyzing layer 3 (i.e., network layer) transport information. *Id.* fig.5. Muller teaches that the address accumulation block 510 can process a packet’s layer 2 header, and the L3 address dependent block 540 can process a packet’s layer 3 header address information, including ports. *Id.* 9:61–10:8. These processes utilize search engine 370, which provides access to the CAM. *Id.* 8:1–12 (describing access to forwarding database 140), 11:7–13 (stating that database 140 includes...
two CAMs). Searching the forwarding database gives instruction on what should be done with the packet. *Id.* 12:47–13:24.

Ariel Hendel, lead inventor on the Hendel patent, appears as a co-inventor on the Muller patent. *Id.* at (75). The Muller and Hendel patents were filed on the same day—June 30, 1997—and have the same assignee. *Id.* at (73), (22).

**Elliott** describes a switching/routing system that uses, inter alia, TCP port information to identify packet flows. Ex. 1019, 18:46–64.

Each proposed Ground is appropriate for institution, as detailed below.4

A. [Ground 1] — Claims 1, 2, 5, 7–10, 12–16, 28, and 29 are anticipated by Feldmeier under 35 U.S.C. § 102

**Claim 1 - [1.0]:** “A method, including the steps of maintaining a set of access control patterns in at least one associative memory:”5

4 The following articulation of grounds incorporates explanations and citations to references that appear in earlier sections of the petition. Moreover, herein, citations to particular portions of references are exemplary and not intended to imply that cited sections are the only portions of the cited-to references that bear on facts supported through citation.

5 The phrase “access control pattern” is not used in the description. It appears in only the claims. Petitioner recognizes that IPR is not an appropriate forum to address failure to comply with 35 U.S.C. § 112 and, therefore, reserves all rights to
Feldmeier describes a “network firewall device” that makes decisions about where to transmit packets, and which packets should be transmitted or discarded. Ex. 1005, 13:7–11; see also id. fig.14 (depicting “network firewall device”). This device utilizes a “network address lookup 1440” that stores information about which packets should be routed to their destinations, and which should be discarded. Id. 13:24–27 (“Demultiplexer 1430 routes the packet received from input packet queue 1410 to either output packet queue 1450 or to discard line 1460 depending on the signal received from network address lookup 1440.”).

Figure 3. Ex. 1005, fig.14 (highlighting added)

Feldmeier teaches the use of CAMs for storing the necessary access control information. Id. 13:19–21 (describing network address lookup 1440 as “an apparatus for performing hierarchical address translation according to an embodiment of the invention”), 5:27–29 (stating, in the “SUMMARY,” that pursue indefiniteness and other infirmities under § 112 in an appropriate forum.)
Feldmeier’s method used “binary or ternary CAMs” for hierarchical address translation), 5:32–35. Feldmeier teaches that the CAM may store network addresses and routing information. *Id.* 5:47–6:8, 5:63–66; *see also* Ex. 1003, ¶¶ 58–60.

In the “network firewall device” embodiment of the Feldmeier patent, these techniques precisely practice the requirements of this limitation. Network lookup 1440 (part of the firewall device) utilizes Feldmeier’s techniques for storing access control information in the CAM, and using information to identify packets to relay or discard. Ex. 1005, 13:19–21. In doing so, Feldmeier teaches “a method, including the steps of maintaining a set of access control patterns in at least one associative memory.”

[1.1] “receiving a packet label responsive to a packet, said packet label being sufficient to perform access control processing for said packet;”
Feldmeier teaches that, upon receipt of a packet, an address extraction module 1420 “extracts a portion of the packet representing an address to which the packet is to be routed.” Ex. 1005, 13:12–17. This portion is then “routed to network address lookup 1440” and used to perform a lookup against information stored therein. Id. 13:15–24. Figure 14 depicts these operations. See also Ex. 1003, ¶¶ 61–62. As demonstrated above, the network address lookup 1440 receives the extracted portion and uses it to perform its lookup, and Feldmeier teaches deriving the packet label from the packet, consistent with patentee Cisco’s application of this term in pending litigation, and thus, consistent with the above-articulated construction of “receiving a packet label responsive to a packet, said packet label being sufficient to perform access control processing for said packet.” See Ex. 1023, at 2 (Cisco’s “responsive” construction).
[1.2] “matching matchable information, said matchable information being responsive to said packet label, with said set of access control patterns in parallel, and generating a set of matches in response thereto, each said match having priority information associated therewith;”

Feldmeier teaches searching the CAM for the address that was extracted from the packet. Figure 9 depicts this step. Feldmeier states that the “comparand” in the search is “the address that is being searched for in the CAM.” Ex. 1005, 10:64–66. As noted supra at [1.0], the CAM in Feldmeier contains access control patterns. Feldmeier teaches that multiple entries in the CAM (i.e., multiple “access control patterns”) may match the comparand, in which case the search may result in a set of multiple matches. Id. 11:10–12 (“If multiple matches with different priorities are found . . . .”); see also id. fig.9 (depicting conditional box 920, labeled “Multiple matching entries?”); Ex. 1003, ¶¶ 63–64.

Figure 5. Ex. 1005, fig.9 (highlighting added)

---

6 “Matchable information” is not used in the ’577 Patent’s written description. It appears in only the claims. Petitioner recognizes that IPR is not an appropriate forum to address failure to comply with 35 U.S.C. § 112 and, therefore, reserves all rights to pursue indefiniteness and other § 112 infirmities in an appropriate forum.
Feldmeier teaches searching the entire CAM at once, i.e., all entries will be searched in parallel. Ex. 1005, 3:28–35 (“CAMs may be implemented using a variety of techniques and technologies. One common technique is to search all CAM entries simultaneously in parallel to find the desired entry.” (emphasis added)). Feldmeier also teaches that each of the matches resulting from searching the CAM for the extracted address will have priority information associated with it. Id. 10:53–54 (“Each time an entry is written into the CAM, both an entry value and a priority value are provided.” (emphasis added)); see also Ex. 1003, ¶ 64.

As demonstrated above, Feldmeier teaches searching the CAM for the address extracted from the packet, with such a search examining all entries in the CAM in parallel, and returning a set of matches, each of which has associated priority values, and Feldmeier teaches this limitation.

[1.3] “selecting at least one of said matches in response to said priority information, and generating an access result in response to said at least one selected match; and”

Feldmeier teaches the use of a priority encoder 1060 “to select the highest priority value [among priority values of matching entries]. Ex. 1005, 11:10–12; see also id. fig.10A, 10B & accompanying disclosure. Once the highest priority value has been identified, Feldmeier teaches re-searching the CAM using both the initial comparand (the extracted address) and the returned highest priority value as inputs. Id. 11:18–29. This search will output “the address of the matching entry of
[the] CAM 1020.” *Id.* 11:20–24. Figure 9 depicts this process, including retrieval of the appropriate entry from the CAM using the output address. *See also id.* 10:51–52 (describing retrieval of entry from the CAM). Feldmeier teaches the use of such information in making access decisions.

Specifically, Feldmeier teaches storing access and routing results in the CAM itself. *Id.* 11:48–49. According to Feldmeier, the CAM search operation generates the access result directly from the CAM itself. *See Ex. 1003, ¶¶ 68.* Alternatively, Feldmeier teaches an address encoder 1090, (id. fig. 10B), which outputs the address of the matching entry. *Id.* 11:21-22; see also Ex. 1003, ¶¶ 65–68. One of ordinary skill would have appreciated that Feldmeier inherently discloses a separate memory that stores the access control and routing results, as the address from the address encoder 1090 strongly implies an additional lookup in a separate memory. *See Ex. 1003, ¶ 68.* It was known in the art to use the address from a CAM lookup to find an associated value—in this case an access control result--in a separate memory, such as an SRAM. *Id.* As demonstrated, Feldmeier teaches the use of priority information to select a match from multiple
matches, and generating an access result in response to that selection, and

Feldmeier teaches this limitation.

[1.4] “making a routing decision in response to said access result.”

Feldmeier teaches that a packet will either be discarded or routed on to a
destination depending on the information received following the CAM lookup. Ex.
1005, 13:24–27 (“Demultiplexer 1430 routes the packet received from input packet
queue 1410 to either output packet queue 1450 or to discard line 1460 depending
on the signal received from network address lookup 1440.”); see also id. fig.14.
Thus, the result received from the network address lookup 1440 governs the
decision where to route the packet. See also Ex. 1003, ¶ 69.

Claim 2 - [2.0] “A method as in claim 1, including the step of performing at
least two of said steps of receiving, matching, selecting, and making a routing
decision, in parallel using a pipeline technique.”

Applying the approach previously described, Feldmeier teaches that the
operations of finding matches to address information in a CAM (“matching”), and

7 Claim 2 is in tension with claim 1. Although claim 1 addresses a sequence of
steps performed serially, with each step’s output serving as a required input for the
next step, claim 2 suggests performing several steps “in parallel using a pipeline
technique.” See Ex. 1003, ¶ 70. Petitioner reserves all right to pursue

indefiniteness and other infirmities under § 112 in an appropriate forum.

21
of identifying the highest priority match (“selecting”), may be performed in a single step via the use of pipelining techniques:

[I]n one embodiment of the present invention, the ternary CAM is **pipelined to provide address resolution in a single cycle** or, if no pipeline is used, in a fixed number of cycles independent of the number of priority levels of the address. This is accomplished by using N bits of the priority field to denote N levels of priority, searching the ternary CAM for the address, decoding the priority fields of the matching entries produced by the search to determine the highest priority field of any matching entries and searching the CAM a second time for the address and the highest priority field of the matching entries. **If the two searches are pipelined, each search requires effectively a single cycle.**

Ex. 1005, 10:16–29 (emphasis added). Feldmeier goes on to describe that such an approach gives an effective search rate of one cycle per operation:

Although a two-cycle operation has been described for simplicity, **the operation can be performed in one cycle.** For example, multiple operations of Fig. 9 can be pipelined so that cycle 1 of a first operation is performed simultaneously with cycle 2 of a second operation, allowing CAM 1000 to perform operations at a rate of one cycle per operation.

*Id.* 11:24–29 (emphasis added); *see also* Ex. 1003, ¶¶ 70–72. In such a design, processing for multiple packets occurs in parallel, with processing for a second packet beginning before processing for a first packet completes. Such is consistent with the ’577 Patent’s discussion of “parallel” processing. Ex. 1001, 3:41–44
(describing multiple elements operating “in parallel,” but not necessarily on the same packet).

As demonstrated above, Feldmeier teaches using pipelining techniques to perform the steps of matching entries, and of selecting the highest priority matches, in parallel, and Feldmeier teaches “a method as in claim 1, including the step of performing at least two of said steps of receiving, matching, selecting, and making a routing decision, in parallel using a pipeline technique.”

**Claim 5 - [5.0]** “A method as in claim 1, wherein said associative memory includes a hardware content-associative memory having a plurality of rows, each row including one of said access control patterns and one of said access results.”

Feldmeier teaches the use of a hardware CAM. Ex. 1005, 3:5–8 (defining “CAM” as a certain type of “memory device”), 5:32–35, 14:18–21 (describing, for clarity, a “hardware implementation”). Feldmeier teaches that the CAM has a plurality of rows. *Id.* fig.8 (depicting multiple rows in a CAM). Feldmeier teaches a CAM in which every row contains a pattern for matching and an instruction as to how a matched packet should be handled. *Id.* (depicting multiple rows); figs.10A–B, 10:66–11:3, 11:48-49.

Feldmeier also teaches using such techniques for making decisions about access control. For instance, in a system for access control, the CAM includes patterns for matching

<table>
<thead>
<tr>
<th>ENTRY</th>
<th>NUMBER</th>
<th>LEVEL</th>
<th>ROUTING DESTINATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>908 979 XXXX</td>
<td>100</td>
<td>STATION NUMBER</td>
</tr>
<tr>
<td>B</td>
<td>908 852 XXXX</td>
<td>100</td>
<td>CENTRAL OFFICE 852</td>
</tr>
<tr>
<td>C</td>
<td>908 XXX XXXX</td>
<td>000</td>
<td>908 TANDEM SWITCH</td>
</tr>
<tr>
<td>D</td>
<td>XXX XXX XXXX</td>
<td>D01</td>
<td>LONG DISTANCE POINT-OF-PRESENCE</td>
</tr>
</tbody>
</table>

*Figure 7. Ex. 1005, fig.8.*
and instructions for whether to drop or route a given packet. *Id.* fig.14 (depicting a system for making such decisions), 13:24–29 (describing that, in figure 14, demultiplexer 1430 receives instructions for whether to route or drop a packet from network address lookup 1440, which includes a CAM using Feldmeier’s technology). Finally, Feldmeier applies the address translation in the context of access control, so as to provide discard or routing result, according to the network firewall device embodiment on Figure 14. *See id.* fig. 14, 13:24–27; *see also* Ex. 1003, ¶¶ 73–74.

As demonstrated above, Feldmeier teaches the use of CAMs that include rows, each of which includes a pattern for matching and information for access control, and Feldmeier teaches “a method as in claim 1, wherein said associative memory includes a hardware content-associative memory having a plurality of rows, each row including one of said access control patterns and one of said access results.”

**Claim 7** - [7.0] “A method as in claim 1, wherein said associative memory includes a ternary content-associative memory.”

Feldmeier teaches the use of a ternary CAM. Ex. 1005, 5:33–36 (“The invention provides methods . . . **using either binary or ternary CAMs**[].”) (emphasis added)), 9:51–10:29 (discussing, in detail, use of ternary CAMs), 6:20–23; *see also* Ex. 1003, ¶ 75.
Claim 8 - [8.0] “A method as in claim 1, wherein said packet label includes a source IP address or subnet, a destination IP address or subnet, a source port, a destination port, a protocol specifier, or an input interface.”

Feldmeier teaches the use of its technology in IP networks. Ex. 1005, 8:31–33 (“Figs. 6A–6F illustrate an example of the [] hierarchical address translation operation of fig. 5 performed on an IP v.4 address.”) (emphasis added)), 12:49–6, fig.13. Feldmeier teaches that a packet received on an IP network includes an “address.” Id. 8:44–45 (discussing “the incoming address”), 13:15–17 (“Address extraction module 1420, in turn, extracts a portion of the packet representing an address to which the packet is to be routed.”); see also fig. 13, 12:61–63. This reference is consistent with conventional IP source and destination address usage, because packets in conventional IP networks include a destination IP address and a source IP address. Ex. 1003, ¶¶ 76–77; see also Ex. 1020, at 168 (depicting conventional fields of an IP packet). Feldmeier thus discloses a “packet label” that includes at least a source IP address and a destination IP address.

Claim 9 - [9.0] “A method as in claim 1, wherein said priority information for each said access control pattern is responsive to a position of said access control pattern in a memory.”

Feldmeier teaches that the priority associated with a CAM entry may be associated with its position within the CAM. Ex. 1005, 3:40–52 (describing that, in some CAMs, it is possible to rely “on the intrinsic priority encoding of entries stored in a CAM”); see also id. 9:54–64. Thus, Feldmeier teaches relying on the
order of CAM entries to determine priority, with the entry latest in the list (i.e., the last) receiving the highest priority treatment, and the entry first in the list the lowest priority treatment. In such a system, entries are stored in the CAM in an order associated with priority. *Id.* 3:40–52; *see also id.* 9:54–64. In Feldmeier’s description, results are returned in “reverse hierarchical order,” so the “first matching entry” will be the one that has been designated—by virtue of its placement last in the list—for highest priority treatment. *Id.* 3:40–52. Feldmeier thus teaches that the first matching entry returned by a search will be the highest-priority, and “also the correct matching entry.” *Id.* In such a system, the priority information is responsive to the position of the entry in the memory. *See also* Ex. 1003, ¶ 78.

**Claim 10 - [10.0] “A method as in claim 1, wherein said priority information includes a position in said associative memory, and said step of selecting includes choosing a first one of said matches.”**

Feldmeier teaches that the priority associated with a CAM entry may be associated with its position within the CAM. Ex. 1005, 3:40–52 (describing that, in some CAMs, it is possible to rely “on the intrinsic priority encoding of entries stored in a CAM”); *see also id.* 9:54–64; *supra* at [9.0]. Thus, Feldmeier teaches relying on the order of CAM entries to determine priority, with the entry latest in the list (i.e., the last) receiving the highest priority treatment, and the entry first in the list the lowest priority treatment. In such a system, entries are stored in the
CAM in an order associated with priority. *Id.* 3:40–52; *see also id.* 9:54–64. In Feldmeier’s description, results are always returned in “reverse hierarchical order,” so the “first matching entry” will be the one that has been designated—by virtue of its placement last in the list—for highest priority treatment. *Id.* 3:40–52.

Feldmeier thus teaches that the first matching entry returned by a search will be the highest-priority, and “also the correct matching entry.” *Id.* In such a system, the priority information is responsive to the position of the entry in the memory, and the selected entry is the first returned match. *See also* Ex. 1003, ¶ 79.

**Claim 12** - [12.0] “A method as in claim 1, wherein said routing decision includes an administrative policy decision regarding treatment of said packet.”

Feldmeier teaches implementing policies set by network administrators, including policies as to which packets should be forwarded and which should be dropped (Ex. 1005, fig.14 & 13:7–29), policies for network load balancing (*id.* figs.15–16 & 13:30–14:12), and policies for general routing (*id.* figs.5–10 & accompanying text). In such uses, decisions about packet handling are administrative policy decisions regarding packet treatment. Ex. 1003, ¶ 80.

**Claim 13** - [13.0] “A method as in claim 1, wherein said routing decision includes determining an output interface for said packet.”

Feldmeier teaches the use of its technology for handling IP packets, and routing (via use of a demultiplexer) received packets to “one of output packet queues 1350, 1360, or 1370 depending on the signal received from IP address
lookup 1340.” Ex. 1005, 13:1–4 & fig.13; see also 13:24–27 (discussing routing received packets to either “output packet queue 1450 or to discard line 1460”) & fig.14. In such uses, the demultiplexer’s decision of how to route the packet includes determining an output interface for the packet, as the demultiplexer will route the packet to the output packet queue identified based on the CAM search. Id., at 13:24–27; see also Ex. 1003, ¶ 81.

**Claim 14 - [14.0]** “A method as in claim 1, wherein said routing decision includes implementing a quality of service policy.”

McAuley (incorporated by reference into Feldmeier, see Ex. 1005, 2:64–3:2) discusses the use of routing as a means to achieve quality of service (“QoS”) objectives. Ex. 1006, at 1383 (discussing a “General Routing Table Lookup Function” that uses QoS information in routing, noting that in some cases, “the QoS information may take priority over the addressing information altogether, for instance because a switch is only concerned about getting the packet to the appropriate IEC [‘Inter-Exchange Carrier’].” (emphasis added)). Giving such considerations precedence implements a quality of service policy. Ex. 1003, ¶ 82.

**Claim 15 - [15.0]** “A method as in claim 1, wherein said routing decision includes permitting or denying access for said packet.”

Feldmeier teaches the use of its technology for determining which packets should be forwarded and which should be dropped. Ex. 1005, fig.14 & 13:7–29 (describing decisions as to whether a given packet should be routed “to either
output packet queue 1450 or to discard line 1460”). Such determinations either permit or deny access, as they either permit the packet to travel on the network or require it to be dropped. Ex. 1003, ¶ 83.

**Claim 16 - [16.0]** “A method as in claim 1, wherein said step of generating said access result is responsive to a plurality of said at least one matches.”

Feldmeier teaches generating access results where more than one entry in the CAM matches the received packet information via prioritizing results. Ex. 1005, 11:9–11 (“If multiple matches with different priorities are found, priority encoder 1060 is used to select the highest priority value.”); see also id. 9:54–64. This matching entry then used to generate the access result, as for example in the Feldmeier’s network firewall embodiment, where the packet is either forwarded or discarded depending on the signal from the IP address lookup 1440. See id. 13:22–27. Such an approach generates an access result responsive to a plurality of matches, as the decision on whether to forward or discard is based on the CAM, for which a plurality of matches is disclosed. Ex. 1003, ¶ 84.

**Claim 28 - [28.0]** “A method as in claim 1, wherein said set of access control patterns is responsive to a sequence of access control specifiers, each one of said sequence of access control specifiers declaring whether to permit or deny access for a set of packets.”

Feldmeier teaches storing strings of bits, each with a match pattern and a mask pattern, in the CAM. Ex. 1005, figs.4A–4B, 6A, 6C–E, 4:19–26 (describing storing in the CAM “binary addresses” and “priority masks”), 8:38–45 (describing
storing the CAM “binary entries” and “CAM masks”). Feldmeier teaches using such CAM entries to determine whether to permit or deny access for incoming packets. *Id.* fig.14 (depicting system for making such decisions), 13:24–29 (describing that, in figure 14, demultiplexer 1430 receives instructions for whether to route or drop a packet from network address lookup 1440, which includes a CAM using Feldmeier’s technology).

Feldmeier further teaches that the binary entries (i.e., “access control patterns”) stored in the CAM are used to enforce rules that describe policies (i.e., “access control specifiers”) for access to network resources. For example, Feldmeier teaches that some packets will be permitted to pass, and others dropped, according to the information in the CAM. *Id.* fig. 14, 13:24–29 (describing how demultiplexer 1430 will route or drop a packet according to the results of the network lookup). *See also* Ex. 1003, ¶ 85.

As demonstrated above, Feldmeier teaches storing in the CAM bit sequences specifying a label match mask and a label match pattern to be used in attempting a match against information from a packet label, and because such bit sequences relate to overall access rules, and Feldmeier teaches this limitation.
Claim 29 - [29.0] “A method as in claim 28, wherein said step of maintaining includes the steps of receiving said sequence of access control specifiers; translating said sequence of access control specifiers into said sequence of access control patterns; and storing said sequence of access control patterns in said associative memory.”

As discussed supra, Feldmeier discusses storing binary entries comprising a label match mask and a label match pattern in the CAM. As discussed supra, Feldmeier discusses using such entries (i.e., “access control patterns”) to enforce rules (i.e., “access control specifiers”) for access control. See supra at [28.0]; see also Ex. 1005, 13:7–11 (describing how a “network firewall device” is used to only allow transmission of certain packets to certain destinations, and to discard packets transmitted to other destinations). That is, the binary entries (i.e., “access control patterns”) give effect to the rules (i.e., “access control specifiers”) and thus the rules “translate” into the bit patterns.8 Feldmeier further teaches that the table

8 The ’577 Patent’s discussion of “translating” contemplates that the term addresses preparing an access control role “for entry into the access control memory.” Ex. 1001, 5:33–35. The ’577 Patent does not describe any specific techniques for such translation, but indicates that the concept covers at least preparing a pattern of bits that reflects the rule in question. Id. at 5:45–53 (describing translation into bit sequences). Consistent with this interpretation, in
setting forth such rules “can be directly stored in a ternary CAM.” Id. 9:51–52. To store such rules in the CAM, it would be necessary to receive the rules to be enforced, then translate the rules into bit strings (i.e., access control patterns) for storage in the CAM, and finally to store the access control patterns in the CAM. Ex. 1003, ¶ 86. This is because a CAM, as a form of electronic storage, stores data only as a string or pattern of bits. See, e.g., Ex. 1003 ¶¶ 41 (describing the various bits involved), 59 (describing the translation of addresses into binary data for CAM storage), 86 (describing translation for this limitation). As the ’577 description acknowledges, ACLs (“access control specifiers”) and techniques for translating, were already known at the time of the alleged invention, and a person of ordinary skill would have appreciated that such would be present inherently in access control devices such as Feldmeier. Ex. 1001, 1:15–2:8 (describing previously-available technologies for access control, including the use of “ACLs (access control lists”), 5:11–38 (describing prior art ACLs in “IOS” software); Ex. 1003, ¶ 86.

related litigation Cisco has confirmed that the “translating” operation includes at least translation into a pattern of bits. Ex. 1023, at 2.
As demonstrated above, Feldmeier inherently teaches receiving sequences of access control specifiers, translating them into access control patterns, and storing them in the CAM, and Feldmeier teaches this limitation.


Hendel describes a network access control device that utilizes a CAM to make access control determinations based on packet header information at layer 2, layer 3, and layer 4 of the OSI layer model. A person of ordinary skill would have appreciated that aspects of the Hendel design would have been highly useful to incorporate into Feldmeier. For example, incorporating Hendel’s techniques for making preliminary routing decisions at both Layer 2 and Layer 3+ would have made the Feldmeier design more flexible and capable, permitting greater degree of management over network traffic and, as a result, better access control. It would have been within the ability of one skilled in the art to combine Hendel and Feldmeier, as discussed further infra. See also Ex. 1003, ¶¶ 88–90.

Claim 18 – [18.0] “A method as in claim 1, wherein said steps of matching and selecting are performed at a rate exceeding 1 megapacket per second.”

Hendel directs its invention to fast networks, including those “reaching to gigabit speeds.” Ex. 1007, 4:12–13. Hendel further directs its invention to packet handling “at wire-speed, i.e., as fast as packets enter the network element.” Id. 4:61–63; see also id. 5:40–42 (repeating “wire-speed” requirement). A person of
skill in the art would have appreciated that packet handling at wire speed, particularly on a network in which transfer speeds reach or even exceed 1 gigabit per second, inherently (and otherwise obviously) required a capability of handling packets at rates exceeding one million packets per second. See Ex. 1003, ¶ 91 (noting that, for gigabit Ethernet, packet forwarding speeds may exceed one million packets per second).

Modifying Feldmeier so as to apply the gigabit speeds taught in Hendel would have increased the Feldmeier device’s speed and utility. Such would have been within the ability of a person of ordinary skill in the art, and as such the proposed combination would have been obvious. See Ex. 1003, ¶ 92.

**Claim 19 - [19.0]** “A method as in claim 1, including the step of making a preliminary routing decision for said packet, wherein said packet routing information includes a result of said preliminary routing decision.”

Hendel teaches making a preliminary routing decision generated at the associated memory 42 for an incoming packet by generating a layer 2 result, i.e. “packet routing information,” that includes the result of a preliminary routing decision (i.e., an output port indicator). Ex. 1007, 11:8–10. This layer 2 result is then used by the merge logic to determine the appropriate final routing determination. Ex. 1007, fig.7 (depicting separate paths for routing lookups at “L2” (layer 2) and “L3” (layer 3), then applying a step of “merge results”), 16:63–
In such a configuration, the layer 2 result including the output port determination, when made, are “preliminary.” See Ex. 1003, ¶ 93.

Modifying Feldmeier so as to utilize the preliminary routing decisions taught in Hendel would have enhanced the Feldmeier device’s speed and efficiency because it would have permitted the Feldmeier device to make L2 routing decisions in parallel with L3 access control decisions. Such would have been within the ability of a person of ordinary skill in the art, and as such the proposed combination would have been obvious. See Ex. 1003, ¶ 94.

Claim 20 - [20.0] “A method as in claim 19, wherein said preliminary routing decision includes determining at least one output interface for said packet.”

Hendel teaches that the layer 2 result corresponding to the preliminary routing decision discussed supra at [19.0] may include determining the “port of the destination address,” i.e., the “output interface.” Ex. 1007, 5:15–19, 11:8–10 (describing determining such port for a layer 2 lookup); see also Ex. 1003, ¶ 95. Modifying Feldmeier so as to determine at least one output interface as part of preliminary routing would have enhanced the Feldmeier device’s speed and efficiency for the reasons discussed supra at [19.0].

Claim 21 - [21.0] “A method as in claim 19, wherein said packet routing information includes an output interface for said packet.”

As discussed supra at [20.0], Hendel teaches determining an output interface for the packet as part of preliminary routing via layer 2 processing. This output
interface information is provided to the merge logic that makes the final forwarding decision, also taking into account the layer 3/layer 4 result. Specifically, Hendel discloses that the layer 2 result determination comprises the output port for the given packet, i.e. “output interface,” (Ex. 1007, 11:8–10), and then this output port information is used by the merge logic to make the final forwarding decision. See id. 13:4–8, 13:65–14:3; see also Ex. 1003, ¶ 96.

Modifying Feldmeier so as to determine an output interface as part of preliminary routing would have enhanced the Feldmeier device’s speed and efficiency for the reasons discussed supra at [19.0].

Claim 22 - [22.0] “A method as in claim 1, including the step of preprocessing said packet label to generate said matchable information.”

Hendel describes a variety of preprocessing steps used to generate the information used to match entries in the CAM. For example, Hendel teaches that class logic 60 examines the packet header information, and uses the results in generating a layer 3 lookup key, i.e., “matchable information.” Ex. 1007, 11:49–67; see also id. fig.4, 10:21–33, 10:55–58; see also Ex. 1003, ¶ 97. Modifying Feldmeier so as to generate the matchable information as part of preliminary routing would have enhanced the Feldmeier device’s speed and efficiency as well as flexibility by expanding their application to various network administrative tasks such as implementing rules based on traffic classification and priority, and for the reasons discussed supra at [19.0].
Claim 25 - [25.0] “A method as in claim 22, wherein said step of preprocessing includes the step of comparing a source IP port value or a destination IP port value with a selected port value.”

Hendel teaches a process of class identification that includes identifying “at least the flow, priority, traffic classification, and hardware routing” of a packet. Ex. 1007, 11:66–67; see also Ex. 1003, ¶ 122. A person of ordinary skill, reviewing Hendel, would have appreciated that identifying a flow requires comparing TCP port numbers. Id. For example, to identify an HTTP flow, the class logic 60 compares the TCP port of an incoming packet to the TCP port number conventionally assigned to HTTP, i.e., port 80. Ex. 1007, 10:42–44 (describing how L3 logic uses information from the class logic to form the search key); see also Ex. 1003, ¶¶ 98–99 (describing same, and confirming HTTP’s conventional port assignment). Having determined that an incoming packet is associated with a known flow type (i.e., HTTP), class logic 60 associates any necessary priority information with the packet. Ex. 1003, ¶¶ 98–99. Class logic 60 then relays that information to L3 logic, where it is used to form a search key. Id. Noting the comparison of the packet’s TCP port (i.e., 80) to a known (i.e., “selected”) port value (i.e., the stored value of “80” denoting an HTTP flow), a person of ordinary skill would conclude that Hendel teaches comparing TCP port values to a selected value. Id.
Modifying Feldmeier so as to examine port information as part of packet preprocessing would have enhanced Feldmeier’s efficiency and flexibility, as it would have permitted additional types of access control policies to be used. See Ex. 1003, ¶ 99. A person of ordinary skill in the art would thus have viewed the proposed combination as obvious. See id.

C. [Ground 3] — Claims 22, 28, 29, and 31 are obvious over Feldmeier in view of Muller under 35 U.S.C. § 103

Claim 22 - [22.0] “A method as in claim 1, including the step of preprocessing said packet label to generate said matchable information.”

Muller further teaches preprocessing a packet header to generate a search key, which is then used as the matchable information in searching the CAM. Ex. 1008, 7:44–66 (describing the operation of header preprocessing logic 305). Specifically, Muller teaches preprocessing that analyzes the packet header for both layer 2 and layer 3 information, and using such information to generate the “search key” that will be used to search the CAM. See also id. 9:56–11:5 (describing the generation of “search
keys” by the blocks in figure 5); 1:63–67, 11:22–35. The “search keys” are matchable information, used to search the CAM. Id. 12:25–26 (“[T]he associative date [i.e., the CAM] is the data with which the search key is matched.”).

Incorporating Muller’s teachings for packet preprocessing into Feldmeier would enhance Feldmeier’s operation and provide useful functionality, such as more capable handling of packets based on layer 2 information, encapsulation information, and layer 3 information. Ex. 1003, ¶¶ 103–04.

Claim 28 - [28.0] “A method as in claim 1, wherein said set of access control patterns is responsive to a sequence of access control specifiers, each one of said sequence of access control specifiers declaring whether to permit or deny access for a set of packets.”

Muller teaches storing access control patterns in a CAM, and confirms that such patterns (i.e., “access control patterns”) are derived from rules (i.e., “access control specifiers”) that govern access to network resources. Ex. 1008, 12:48– 13:24 (stating that the data in the CAM lists ports “to which the packet may be forwarded,” as well as “priority fields for priority tagging and priority queuing,” and “a best effort mask indicating which ports should queue the packet as a best effort”). A person of skill would have readily appreciated that such a system could also store patterns (i.e., “access control patterns”) derived from rules (i.e., “access control specifiers”) governing packet access to network resources, as set forth in Feldmeier’s discussion of firewalls. See supra Feldmeier [28.0, 29.0]; see also Ex. 1003, ¶¶ 105–06.
A person of skill would have had ample incentive and ability to combine Muller with Feldmeier in the proposed manner, as such would complement and enhance Feldmeier’s functionality and confer additional features, such as more capable handling of packets based on layer 2 information, encapsulation information, and layer 3 information. Ex. 1003, ¶¶ 105–06. Muller’s teachings for deriving access control patterns from rules that recite filtering functions (i.e., permitting or denying access) reinforce the utility of Feldmeier’s network address lookup for access control. Ex. 1005, 13:19–21; Ex. 1003, ¶¶ 107–08.

**Claim 29 - [29.0] “A method as in claim 28, wherein said step of maintaining includes the steps of receiving said sequence of access control specifiers; translating said sequence of access control specifiers into said sequence of access control patterns; and storing said sequence of access control patterns in said associative memory.”**

Muller further teaches the additional limitations of claim 29. In order to properly store the access control patterns described *supra* regarding Muller [28.0], it would be necessary, and otherwise obvious, to receive the rules to be enforced (i.e., “access control specifiers”), then translate the rules into bit strings for storage in the CAM (i.e., the “access control patterns”), and finally to store the access control patterns in the CAM. See Ex. 1001, 5:11–24 (describing the prior art Cisco access control list syntax, written in human-readable language); 5:33–38 (describing how access control entries ultimately become bit-strings, or access control patterns, stored in the CAM.) This is because a CAM, as a form of
electronic storage, stores data only as a string or pattern of bits. See Ex. 1003 ¶¶ 41 (describing the various bits involved), 59 (describing the translation of addresses into binary data), 107 (describing translation for this limitation); see also supra Feldmeier [29.0] n.8. A person of ordinary skill would have readily understood this from the disclosure of Muller. Ex. 1003, ¶¶ 107–08. A person of skill would have had ample incentive and ability to combine Muller with Feldmeier in the proposed manner, as such would complement and enhance Feldmeier’s functionality and confer additional features, such as more capable handling of packets based on layer 2 information, encapsulation information, and layer 3 information. See supra [28.0]; Ex. 1003, ¶¶ 105–06. Muller’s teachings for deriving access control patterns from rules that recite filtering functions (i.e., permitting or denying access) reinforce the utility of Feldmeier’s network address lookup for access control. Ex. 1005, 13:19–21; Ex. 1003, ¶¶ 107–08.

Claim 31 - [31.0] “A method as in claim 29, wherein said step of translating includes the step of generating a single one of said access control patterns in response to a plurality of said access control specifiers.”

Like Feldmeier, Muller further teaches storing a single pattern in the CAM for multiple network access rules. Indeed, Muller teaches the use of variable (i.e., wildcard) bits in the CAM. Ex. 1008, 12:45 (noting that the CAM may be a mask per bit CAM supporting such bits). Applying wildcard bits in the manner disclosed, a person of ordinary skill in the art would have appreciated that it was
inherent, and otherwise obvious, that a CAM address location storing a single bit string (i.e., “access control pattern”) could represent multiple rules (i.e., “access control specifiers”). That is, multiple access control rules having a complementary effect (e.g., multiple rules that together require the same treatment for all telephone numbers in a single area code) may translate into a single bit-string when used with wildcards that mask out the parts of the bit-strings that differ (e.g., the numbers besides the area code). Rather than using one CAM entry for each of the above-described rules, a person of ordinary skill would readily appreciate that a single bit string using wildcards could achieve the same purpose, i.e., it could implement multiple access rules. Ex. 1005, 2:16–36 (describing how, by using “don’t care” bits, multiple telephone numbers receive the same routing treatment). Indeed, taking disclosures of both Muller and Feldmeier into account, it would have been obvious to a person of ordinary skill in the art that the wildcard functionality in both references was used to translate multiple access control rules (i.e., “access control specifiers”) into a single bit string to be stored in a CAM (i.e., “access control pattern”). Ex. 1003, ¶¶ 109–10. Such an adaptation would have been within the ability of such a person, and would have resulted in increased efficiency (particularly valued since CAM storage space was known to be expensive) and capability (permitting implementation of a greater number of rules in the same amount of CAM space). *Id.*
D. [Ground 4] — Claim 30 is obvious over Feldmeier in view of Muller and the knowledge of one of ordinary skill in the art under 35 U.S.C. § 103

Claim 30 - [30.0] “A method as in claim 29, wherein said step of translating includes the step of generating a plurality of said access control patterns in response to one of said access control specifiers.”

As discussed supra at Feldmeier [29.0] and Feldmeier-Muller [29.0], a person of ordinary skill in the art would have appreciated, from reviewing those references, the need to create a pattern of bits (i.e., “access control pattern”) from some rule (i.e., “access control specifier”). Such a person would also have understood, or have viewed it as obvious, that the techniques necessarily involved in such a process would in certain circumstances require multiple bit patterns to fully implement corresponding single rules. This is because, as the ’577 Patent itself acknowledges, it was well-known at the time of the alleged invention that implementing rules utilizing address or port ranges requires separate binary expressions to test each of the binary “bits” implicated by the range. Ex. 1001, 5:57–64 (in a “common” port comparison, requiring testing “each one of the six high-order bits of the port number for being logical ‘1,’” and so requiring “about six entries” in the CAM”); Ex. 1003, ¶¶ 113–15. The reason for this stems from the nature of binary expressions; a port or address range that is expressible in simple terms using decimal integers is often more complex when reduced to a
sequence of binary expressions sufficient to fully capture the entire range. Ex. 1003, ¶¶ 111–17.

As previously discussed, Feldmeier and Muller teach and/or render obvious claim 29’s requirement of translating access control specifiers into access control patterns. In view of the above, and in view of Feldmeier and Muller’s teachings concerning the translation of access control specifiers into access control patterns, a person of ordinary skill would have viewed it is intrinsic, and otherwise obvious, that implementing certain rules according to the guidance of Feldmeier and Muller would require generating multiple “access control patterns” from a single “access control specifier.”


**Claim 25** - [25.0] “A method as in claim 22, wherein said step of preprocessing includes the step of comparing a source IP port value or a destination IP port value with a selected port value.”

Elliott further teaches a packet classifier that involves comparing TCP port values (i.e., source IP port values and/or destination IP port values) to stored values, Ex. 1019, 18:46–64 (describing packet classifier 285, corresponding, in combination with Muller’s L3 header class matching block 530, L3 address dependent block 540, and search engine 370, to ’853 Patent comparison circuit 230 and CAM 232). Elliott describes keeping “a table of flow specifications,” including TCP port numbers. *Id.* 18:46–64. To determine, as Elliott describes, “if
[a] packet belongs to any special prioritized group” by using such port numbers means comparing the port number on the incoming packet to the “selected” (i.e., previously entered into the table) port value. *Id; see also* Ex. 1003, ¶ 120.

One of ordinary skill in the art would have found it obvious to incorporate Elliott’s port comparison techniques and Muller’s preprocessing logic into Feldmeier, as both references are highly similar in subject matter and the combination would bring substantial benefits in packet classification and prioritizing. *See also* Ex. 1003, ¶¶ 120–21. Thus, Feldmeier in view of Muller and Elliott renders obvious equivalent functions and structure (e.g., Muller’s L3 header class matching block 530, L3 address dependent block 540, and search engine 370, and Elliott’s packet classifier, corresponding to the ’853 Patent’s compare circuit 230 and CAM 232) practicing the limitations of claim 56.

Modifying Feldmeier in the proposed manner would confer operational flexibility, including the implementation of access control rules with based on traffic classification and priority, and as such the proposed combination would have been obvious to one of ordinary skill in the art. *See* Ex. 1003, ¶ 120–21.

**VII. THIS PETITION DOES NOT PRESENT REDUNDANT GROUNDS FOR INSTITUTION**

On April 10, 2015, Petitioner filed a Petition for IPR of the ’577 Patent, PTAB Dkt. No. IPR2015-00973. This Petition presents further, non-redundant grounds for IPR. Hendel and Feldmeier emphasize different features, each relating
to access control and aspects of the claims addressed above. Hendel (the primary reference in the -973 Petition) teaches multi-layer access control. Parallel processing techniques simultaneously evaluate network traffic for access control at L2, L3, and even L4 and above. Hendel is particularly strong for those claims requiring parallel processing (e.g., claim 2) and packet preprocessing (e.g., claims 22, 25). Hendel’s multi-layer approach to packet handling is also relevant to the “routing decision” claims (e.g. claims 1, 12-15), and especially claim 14 for quality of service routing. Finally, Hendel describes the need for high speeds, addressing claim 18’s requirement of routing speeds exceeding one megapacket/second.

Feldmeier (primary reference in the instant petition) provides depth with respect to different aspects of the Challenged Claims. Feldmeier’s extensive discussion of a “priority encoder,” and that encoder’s role in network routing decisions, is highly relevant to claim 1’s priority-related requirements, which are reflected in all Challenged Claims. Feldmeier also includes a robust discussion of packet filtering, describing in detail how certain packets are permitted, and others denied, entry into a given network. While permit and deny decisions are also discussed in Hendel, Feldmeier’s discussion is robust, and is buttressed by figures and text. Feldmeier also offers extensive discussion of the masks and “don’t care” bits that are an aspect of ternary content-associative memories (see claim 7). Finally, Feldmeier gives a detailed discussion of the translation considerations
attending translating rules into bit strings for storage in the CAM, as addressed in claims 28–31.

The grounds under § 103 are not redundant of each other or of the § 102 grounds. The obviousness references each disclose specific implementation details that add new approaches and features to those described in Hendel.

**VIII. CONCLUSION**

The references identified in this Petition indicate a reasonable likelihood of success as to Petitioner’s assertion that the Challenged Claims are not patentable pursuant to the grounds presented. Accordingly, Petitioner respectfully requests institution of IPR for the Challenged Claims for each of the grounds presented.
Respectfully submitted,

Dated: April 16, 2015

/W. Karl Renner/
W. Karl Renner, Reg. No. 41,265
Kevin E. Greene, Reg. No. 46,031
Fish & Richardson P.C.
P.O. Box 1022
Minneapolis, MN 55440-1022
T: 202-783-5070
F: 202-783-2331
Attorneys for Petitioner
CERTIFICATE OF SERVICE

Pursuant to 37 CFR §§ 42.6(e)(4)(i) et seq. and 42.105(b), the undersigned certifies that on April 16, 2015, a complete and entire copy of this Petition for Inter Partes Review and all supporting exhibits were provided via FedEx to the Patent Owner by serving the correspondence address of record as follows:

Campbell Stephenson LLP
11404 Century Oaks Terrace
Bldg. H, Suite 250
Austin, TX 78758

/Susan Johnson/

Susan Johnson
Fish & Richardson P.C.
3200 RBC Plaza
60 South Sixth Street
Minneapolis, MN 55402
(214) 292-4086