

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner,

v.

LIONRA TECHNOLOGIES LIMITED,
Patent Owner.

IPR2024-01281
Patent 7,738,471 B2

Before WILLIAM V. SAINDON, STEVEN M. AMUNDSON, and
RUSSELL E. CASS, *Administrative Patent Judges*.

AMUNDSON, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)

I. INTRODUCTION

Cisco Systems, Inc. (“Petitioner”) filed a Petition requesting an *inter partes* review of claims 1–3, 5–10, 12–15, and 17–21 in U.S. Patent No. 7,738,471 B2 (“the ’471 patent”) (Exhibit 1001) under 35 U.S.C. §§ 311–319. Paper 2 (“Pet.”). Lionra Technologies Limited (“Patent Owner”) did not file a Preliminary Response.

In the Institution Decision, we instituted review based on all challenged claims and all challenges included in the Petition. Paper 6 (“Inst. Dec.”).

We have jurisdiction under 35 U.S.C. § 6. We issue this Final Written Decision under 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons explained below, Petitioner has shown by a preponderance of the evidence that claims 1–3, 5–10, 12–15, and 17–21 in the ’471 patent are unpatentable. *See* 35 U.S.C. § 316(e).

II. BACKGROUND

A. Procedural History

After we instituted review, Patent Owner filed a Response to the Petition. Paper 12 (“Resp.”). Petitioner filed a Reply to Patent Owner’s Response. Paper 15 (“Reply”). Patent Owner filed a Sur-reply to Petitioner’s Reply. Paper 18 (“Sur-reply”).

In December 2025, we held an oral hearing. The record includes the hearing transcript. Paper 25 (“Tr.”).

B. Real Parties in Interest

Petitioner identifies itself as the sole real party in interest. Pet. 72. Patent Owner identifies itself as the sole real party in interest. Paper 4, 2. The parties do not raise any issue about real parties in interest.

C. Related Matters

Petitioner and Patent Owner identify the following civil action as a related matter involving the '471 patent: *Lionra Technologies Ltd. v. Cisco Systems, Inc.*, No. 2:24-cv-00097 (E.D. Tex. filed Feb. 13, 2024) (the “Texas case”). Pet. 72; Paper 4, 2.

D. The '471 Patent (Exhibit 1001)

The '471 patent, titled “High Speed Packet Processing in a Wireless Network,” issued on June 15, 2010, from an application filed on September 14, 2007. Ex. 1001, codes (22), (45), (54). The patent states that the invention “relates to a method and apparatus for high speed protocol header processing at an intermediate and/or destination node of a packet-based communications network.” *Id.* at 1:8–11.

The '471 patent defines a packet-based communications network as “an internet protocol (IP) based communications network [or] a transmission control protocol/internet protocol (TCP/IP) based communications network.” Ex. 1001, 1:15–19, 4:58–62.

The '471 patent describes a “header processing approach” for transmitting protocol data and/or application data that purports to “improve the abilities” of packet-based communications networks. Ex. 1001, 1:19–21. “To enable the transmission of packets between a set of nodes, the application data is encapsulated between one or more protocol stack layer headers and a frame check sequence (FCS).” *Id.* at 1:30–33. The “encapsulation framework depends on the network’s protocol model, such as an open system interconnection [OSI] model or a Department of Defense [DoD] protocol model.” *Id.* at 1:56–59; *see id.* at 6:59–63.

The '471 patent describes one common encapsulation framework as follows:

For example, the application data is encapsulated between an application layer header (or protocol header) and the frame check sequence (FCS). The application layer header is encapsulated between a transport layer protocol header and the application data. The transport layer protocol header is encapsulated between a network layer protocol header and the application layer header. The network layer protocol header is encapsulated between a logic link control (LLC) layer protocol header and the transport layer protocol header. The logic link control (LLC) layer protocol header is encapsulated between a media access control (MAC) layer protocol header and the network layer protocol header. The media access control (MAC) layer protocol header is encapsulated between a physical layer protocol header and the logic link control (LLC) protocol header. The physical layer protocol header is encapsulated between a preamble and the media access control (MAC) layer protocol header.

Ex. 1001, 1:40–56; *see id.* at 8:22–40.

The '471 patent's Figure 3 (reproduced below) depicts a packet implementing an encapsulation framework:

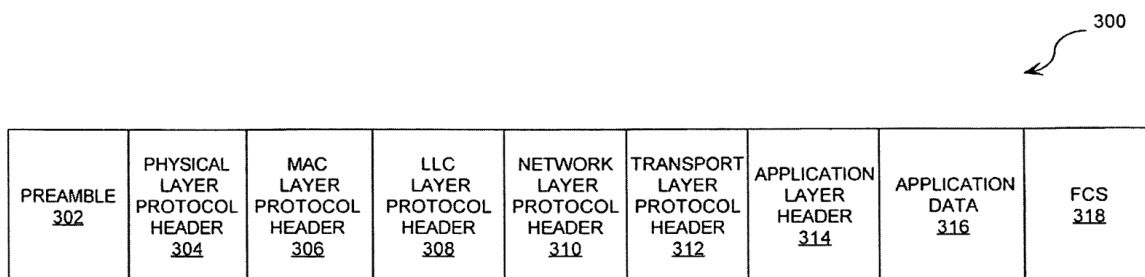


FIG. 3
(PRIOR ART)

Figure 3 illustrates packet 300 comprising “a preamble 302, a physical layer protocol header 304, a media access control (MAC) layer protocol header 306, a logic link control (LLC) layer protocol header 308, a network layer protocol header 310, and a transport layer protocol header 312” as well as “an application layer header 314, an application data 316, and a frame check sequence (FCS) 318.” Ex. 1001, 8:1–9, Fig. 3.

The ’471 patent discusses drawbacks with sequentially processing encapsulated headers, such as (1) the inability to update “header fields” and (2) the inability to “operate at a desired speed for a battery powered wireless communications device.” *See* Ex. 1001, 2:14–56. The patent identifies “a need for a solution to reduce protocol header processing time” in packet-based communications networks, including providing (1) the ability to update “header fields” and (2) the ability to “process protocol headers at a desired speed for a battery powered wireless communications device.” *Id.* at 2:57–62.

The ’471 patent endeavors to address that need by disclosing “a method for processing a packet at an egress end user node” (EEUN) including a “decoding” step and a “concurrent writing” step. Ex. 1001, 3:6–8; *see id.* at code (57). The “decoding step involves decoding a packet having a plurality of headers.” *Id.* at 3:8–9; *see id.* at code (57). The “concurrent writing step is performed subsequent to the decoding step.” *Id.* at 3:9–11; *see id.* at code (57). The “concurrent writing step involves concurrently writing (1) each of the headers to a packet buffer memory and (2) each individual one of the headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.” *Id.* at 3:11–15; *see id.* at code (57).

According to an “aspect of the invention,” the “method includes concurrently processing the headers in each of a plurality of protocol stack layers.” Ex. 1001, 3:32–34. The “method also includes communicating a portion of the packet to a direct memory access (DMA) device subsequent to the decoding step and prior to the concurrent writing step.” *Id.* at 3:34–37. The “portion of the packet is absent of a preamble and a physical layer header.” *Id.* at 3:37–38. The “method further includes communicating a physical layer header to a physical layer memory where it is available for immediate processing within a physical layer.” *Id.* at 3:38–41.

The ’471 patent also discloses an “egress end user node (EEUN)” including a decoder and a DMA device coupled to the decoder. Ex. 1001, 3:45–47, 3:49. The “decoder is configured to decode a packet having a plurality of headers.” *Id.* at 3:47–49. The “DMA device is configured to concurrently write (1) each of the headers to a packet buffer memory and (2) each individual one of the headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.” *Id.* at 3:49–54; *see id.* at 5:64–6:1, 6:38–43.

The '471 patent's Figure 1 (reproduced below) depicts a packet-based communications system:

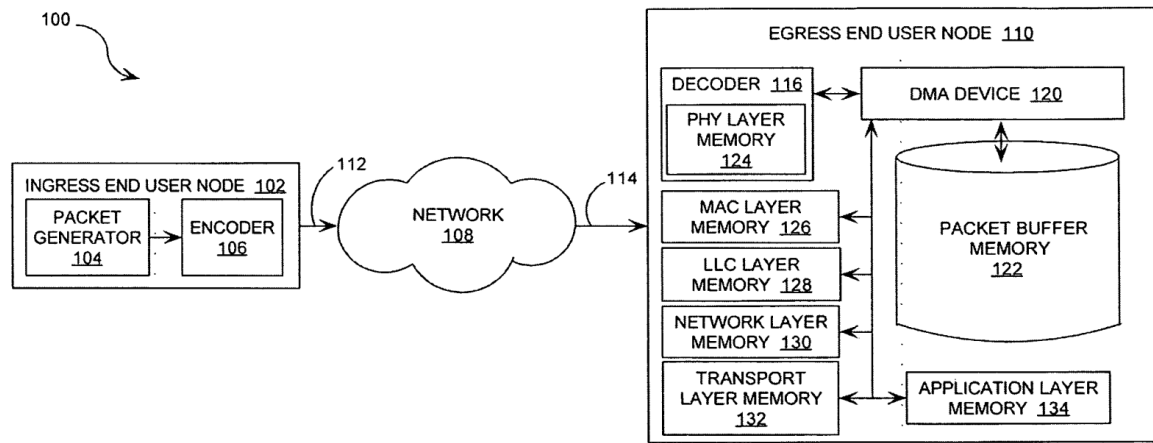


FIG. 1

Figure 1 illustrates packet-based communications system 100 comprising “an ingress end user node (IEUN) 102, a network 108, and an egress end user node (EEUN) 110.” Ex. 1001, 4:20–22, 4:38–41, Fig. 1.

IEUN 102 “is comprised of a packet generator 104 and an encoder 106.” Ex. 1001, 4:47–48, Fig. 1. IEUN 102 “is a source of a plurality of packets including data packets and signaling packets.” *Id.* at 4:41–43.

EEUN 110 “is comprised of a decoder 116, a direct memory access (DMA) device 120, and a packet buffer memory 122” as well as “a media access control (MAC) layer memory 126, a logic link control (LLC) layer memory 128, a network layer memory 130, a transport layer memory 132, and an application layer memory 134.” Ex. 1001, 5:3–9, Fig. 1.

Decoder 116 “is comprised of a physical layer memory 124” for “storing a portion of a decoded packet including preamble and physical layer information.” Ex. 1001, 5:15–18, Fig. 1; *see id.* at 9:22–28. Decoder 116 “advantageously forward[s] a portion of a decoded packet that is absent of

preamble and physical layer information to the DMA device 120.” *Id.* at 5:18–21; *see id.* at 9:28–40.

DMA device 120 “is configured to process a received portion of a decoded packet.” Ex. 1001, 6:7–8. DMA device 120 “concurrently communicates data to the different memories,” i.e., to “the packet buffer memory 122, the media access control (MAC) layer memory 126, the logic link control (LLC) layer memory 128, the network layer memory 130, the transport layer memory 132, and the application layer memory 134.” *Id.* at 6:14–19, 6:22–24, Fig. 1; *see id.* at 9:41–59, 9:65–10:4. DMA device 120 “allows the transfer of data to these various memory locations in a single DMA transaction.” *Id.* at 6:20–21.

For example, DMA device 120 may (1) “temporarily store headers from a MAC layer, the LLC layer, the network layer, the transport layer and an application layer in a memory internal to the DMA device 120” and (2) “temporarily store payload data in a memory internal to the DMA device 120.” Ex. 1001, 6:32–38. After temporarily storing the headers and the payload data, DMA device 120 “performs actions to concurrently write the headers and payload to the packet buffer memory 122 and at least one of the MAC layer memory 126, the LLC layer memory 128, the network layer memory 130, the transport layer memory 132, and the application layer memory 134.” *Id.* at 6:38–43.

According to the ’471 patent, concurrently communicating data to the different memories achieves the following advantages:

- allows for “immediate processing”;
- “facilitates a high data transfer rate by allowing the exchange of information between the protocol stack layers”; and

- “reduces processing latency and thereby increases bandwidth.”

Ex. 1001, 5:64–67, 6:2–6; *see id.* at 6:28–31, 10:9–17.

E. The Challenged Claims

The ’471 patent includes 22 claims. Ex. 1001, 11:51–14:5. Petitioner challenges the following claims:

- independent claim 1 for a “method for processing a packet at an egress end user node”;
- claims 2, 3, 5–10, and 12 that depend directly or indirectly from claim 1;
- independent claim 13 for an “egress end user node (EEUN) of a packet based communications system”; and
- claims 14, 15, and 17–21 that depend directly or indirectly from claim 13.

Pet. 7, 16–69.

Claims 1 and 13 exemplify the challenged claims and read as follows (with formatting added for clarity and with bracketed numbers added for reference purposes):¹

1. [1.0] A method for processing a packet at an egress end user node, comprising:
 - [1.1] decoding a packet having a plurality of headers; and
 - [1.2] subsequent to said decoding step, concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.

¹ We use the same reference numbers that Petitioner uses to identify the claim language. *See* Pet. 32–45, 55–56.

13. [13.0] An egress end user node (EEUN) of a packet based communications system, comprising:

[13.1] a decoder configured for decoding a packet having a plurality of headers; and

[13.2] a direct memory access (DMA) device coupled to said decoder and

[13.3] [the DMA device] configured for concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.

Ex. 1001, 11:51–59, 12:33–43.

F. The Asserted References

For its challenges, Petitioner relies on the following references:

Name	Reference	Exhibit
Cornett	US 2006/0072564 A1, published April 6, 2006 (based on an application filed May 26, 2005)	1005
Nelson	US 7,333,489 B1, issued February 19, 2008 (based on an application filed October 24, 2000)	1006
Paatela	US 2006/0209840 A1, published September 21, 2006 (based on an application filed September 12, 2005)	1007
Russell	US 6,678,746 B1, issued January 13, 2004 (based on an application filed August 1, 2000)	1025

Pet. 16–69.

Petitioner contends that:

- (1) Cornett and Russell qualify as prior art under § 102(b);
and
- (2) Nelson and Paatela qualify as prior art under § 102(e).

Pet. 16; *see* 35 U.S.C. § 102(b), (e) (2006).²

Patent Owner does not dispute that Cornett, Nelson, Paatela, and Russell qualify as prior art. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

G. The Asserted Challenges to Patentability

Petitioner asserts the following challenges to patentability:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
1–3, 5–10, 12–15, 17–21	103(a)	Cornett, Paatela, Nelson
8	103(a)	Cornett, Paatela, Nelson, Russell

Pet. 16–69.

H. Testimonial Evidence

To support its challenges, Petitioner relies on the Declaration of Nader F. Mir, Ph.D. (Exhibit 1003) submitted with the Petition. Dr. Mir states that he has been “asked to provide [his] opinions regarding whether the subject matter of claims 1–3, 5–10, 12–15, [and] 17–21” in “the ’471 patent would have been obvious to a person having ordinary skill in the art” at “the time of the alleged invention, in light of the prior art.” Ex. 1003 ¶ 3. Dr. Mir also states that he received (1) “a B.S. in electrical engineering from Polytechnic University” in 1985, (2) “an M.S. degree in electrical engineering in 1990” from Washington University, and (3) “a Ph.D. in electrical engineering from Washington University” in 1995. *Id.* ¶ 9.

² The Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112-29, 125 Stat. 284 (2011), amended 35 U.S.C. § 102 and § 103 effective March 16, 2013. Because the filing date of the challenged claims predates the AIA’s amendments to § 102 and § 103, this decision refers to the pre-AIA versions of § 102 and § 103. But our patentability analysis would not change if the current versions of § 102 and § 103 applied.

To support its positions, Patent Owner relies on the Declaration of Thomas Brooks (Exhibit 2005) submitted with the Response. Mr. Brooks states that Patent Owner “has retained me as an independent expert in connection with IPR2024-01281 to provide testimony regarding the technology involved in” the ’471 patent. Ex. 2005 ¶ 1. Mr. Brooks also states that he “received a Bachelor of Science degree in computer engineering from the Virginia Polytechnic Institute and State University” in 1998 and has “worked extensively in the design of computer networking and computer systems, including both in hardware design and software design.” *Id.* ¶ 4.

Additionally, Petitioner submits Mr. Brooks’s deposition testimony. *See* Ex. 1027 (Aug. 26, 2025, Brooks Dep. Tr.).

I. Burden

In an *inter partes* review, a petitioner bears the burden of persuasion to prove unpatentability by “a preponderance of the evidence.” *Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (quoting 35 U.S.C. § 316(e)); *see* 37 C.F.R. § 42.1(d) (2025).

III. PATENTABILITY ANALYSIS

A. Legal Principles: Obviousness

A patent may not be obtained “if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” 35 U.S.C. § 103(a) (2006). An obviousness analysis involves underlying factual inquiries including (1) the scope and content of the prior art; (2) differences between the claimed invention and the prior art; (3) the level

of ordinary skill in the art; and (4) where in evidence, objective indicia of nonobviousness, such as commercial success, long-felt but unsolved needs, and failure of others.³ *Graham v. John Deere Co.*, 383 U.S. 1, 17–18, 35–36 (1966); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1047–48 (Fed. Cir. 2016) (en banc). When evaluating a combination of references, an obviousness analysis should address “whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007).

We analyze the obviousness issues according to these principles.

B. Level of Ordinary Skill in the Art

Factors pertinent to determining the level of ordinary skill in the art include (1) the educational level of the inventor; (2) the type of problems encountered in the art; (3) prior-art solutions to those problems; (4) the rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the educational level of workers active in the field. *Envtl. Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 696–97 (Fed. Cir. 1983). Not all factors may exist in every case, and one or more of these or other factors may predominate in a particular case. *Id.* These factors are not exhaustive, but merely a guide to determining the level of ordinary skill in the art. *Daiichi Sankyo Co. v. Apotex, Inc.*, 501 F.3d 1254, 1256 (Fed. Cir. 2007). Moreover, the prior art itself may reflect an appropriate skill level. *Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

³ The parties do not present evidence or argument about objective indicia of nonobviousness. *See, e.g.*, Pet. 17–69; Resp. 18–38; Reply 8–23; Sur-reply 7–21.

Petitioner asserts that a person of ordinary skill in the art at the time of the alleged invention would have had:

- (1) “a bachelor’s degree in computer science, computer engineering, electrical engineering, or an equivalent training”;
- (2) “approximately two years of work experience in network communications”; and
- (3) “a working knowledge of network communications techniques and high-speed packet processing technologies.”

Pet. 13. Petitioner also asserts that “[l]ack of work experience can be remedied by additional education, and vice versa.” *Id.* Dr. Mir’s testimony supports Petitioner’s assertions. *See* Ex. 1003 ¶ 20.

Patent Owner asserts that a person of ordinary skill in the art would have had either “a Bachelor’s degree in computer science, computer engineering, or electrical engineering and at least two years of experience in computer networking, including packet-based networking such as the Internet Protocol,” or “equivalent work experience in the design and/or implementation of systems involving computer networking.” Resp. 8. Patent Owner also asserts that “[a]dditional education in the field could be substituted for industry experience, and vice versa.” *Id.* at 8–9. Mr. Brooks’s testimony supports Patent Owner’s assertions. *See* Ex. 2005 ¶ 14.

Regarding an ordinarily skilled artisan’s education, the parties generally agree on the level of education (“bachelor’s degree”) and the field of study (“computer science, computer engineering, or electrical engineering”). *See* Pet. 13; Resp. 8. Regarding an ordinarily skilled artisan’s experience, the parties generally agree on the area of experience

(“network communications”/“computer networking”), but Patent Owner’s description allows for more experience (“at least two years”) than Petitioner’s description (“approximately two years”). Pet. 13; Resp. 8.

We adopt Petitioner’s description of an ordinarily skilled artisan because it comports with the technology and claims in the ’471 patent as well as the asserted references. *See, e.g.*, Ex. 1001, 1:7–4:13, 11:51–14:5; *infra* §§ III.D.1–III.D.3, III.E.1. We note that our analysis would not change if we adopted Patent Owner’s description of an ordinarily skilled artisan, which allows for more experience (“at least two years”) than Petitioner’s description (“approximately two years”). *See* Pet. 13; Resp. 8. Someone with a higher skill level will understand the technology at least as well as someone with a lower skill level. *See Innovention Toys, LLC v. MGA Ent., Inc.*, 637 F.3d 1314, 1323 (Fed. Cir. 2011); *Tokai Corp. v. Easton Enters., Inc.*, 632 F.3d 1358, 1369 (Fed. Cir. 2011). Hence, “it is generally easier to establish obviousness under a higher level of ordinary skill.” *Kinetic Concepts, Inc. v. Smith & Nephew, Inc.*, 688 F.3d 1342, 1366 (Fed. Cir. 2012).

C. Claim Construction

1. GENERAL PRINCIPLES

We construe claim terms “using the same claim construction standard” that district courts use to construe claim terms in civil actions for patent infringement. *See* 37 C.F.R. § 42.100(b). Under that standard, claim terms “are given their ordinary and customary meaning, which is the meaning the term would have to a person of ordinary skill in the art at the time of the invention.” *Power Integrations, Inc. v. Fairchild Semiconductor Int’l, Inc.*, 904 F.3d 965, 971 (Fed. Cir. 2018) (citing *Phillips v. AWH Corp.*,

415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc)). The meaning of claim terms may be determined by “look[ing] principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17).

2. PETITIONER’S POSITION

In the Petition, Petitioner “contends that no claim term requires express construction” for “the purposes of this proceeding.” Pet. 13 (citing Ex. 1003 ¶ 41).

In the Reply, Petitioner addresses the following language in each of limitations 1.2 and 13.3: “concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers to a respective protocol stack layer memory” (together called the “concurrently writing” requirement). *See* Reply 7–8. Petitioner notes that the district court in the Texas case adopted the following construction for the “concurrently writing” requirement:

Plain meaning, i.e., “concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.”

Id. at 7 (quoting Ex. 2002 (Memorandum Opinion and Order), 8).

Petitioner contends that if the Board construes the “concurrently writing” requirement, the Board should adopt the same construction as the district court in the Texas case. Reply 8.

3. PATENT OWNER’S POSITION

In the Response, Patent Owner urges the Board to adopt the same construction for the “concurrently writing” requirement as the district court in the Texas case. Resp. 12; *see* Sur-reply 24. Patent Owner asserts that Petitioner and Patent Owner agreed to that construction in the Texas case. Resp. 12. According to Patent Owner, Petitioner considers the district court’s construction “dispositive and has provided no explanation of why this construction should not be applicable in this *inter partes* review.” *Id.* at 13; Sur-reply 4–5. Patent Owner also asserts that the claim language, the ’471 patent’s written description, and the ’471 patent’s prosecution history confirm that construction. *See* Resp. 14–18; Sur-reply 6.

According to Patent Owner, the “concurrently writing” requirement means that:

- (1) “each of said plurality of headers” must be written “to a packet buffer memory” and “to a respective protocol stack layer memory” concurrently; and
- (2) “each individual one of said plurality of headers” must be written to each “respective protocol stack layer memory” concurrently.

Resp. 10–11.

Patent Owner does not propose an express construction for any other claim term, phrase, or limitation. *See, e.g.*, Resp. 9–38; Sur-reply 3–21.

4. DISCUSSION

(a) The “Concurrently Writing” Requirement

Based on the claim language and the ’471 patent’s specification, we agree with the parties and with the district court in the Texas case that the “concurrently writing” requirement means “concurrently writing (1) each of

said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.” *See* Resp. 12; Reply 7–8; Sur-reply 1, 4–6; Ex. 1001, 3:9–15, 3:49–54, 5:64–6:1, 6:14–31, 6:38–43, 9:41–59, 9:65–10:4, 11:54–59, 12:37–43, code (57); Ex. 2002, 8.

Regarding the claim language, for example, the phrase “concurrently writing” modifies both item (1) and item (2) that follow the phrase. Ex. 1001, 11:54–59, 12:38–42. As Petitioner asserts and Patent Owner agrees, “[m]iddle-school grammar rules dictate that a modifier before a numbered list modifies each item in the list.” Resp. 14; Sur-reply 5–6; Ex. 2004,⁴ 1; *see SIMO Holdings Inc. v. Hong Kong uCloudlink Network Tech. Ltd.*, 983 F.3d 1367, 1377 (Fed. Cir. 2021) (citing Antonin Scalia & Bryan A. Garner, *Reading Law: The Interpretation of Legal Texts* § 19, 147 (2012)). So the phrase “concurrently writing” applies to “both items in the numbered list.” *See* Ex. 1001, 11:54–59, 12:38–42; Ex. 2004, 7; Resp. 14–15.

Consistent with the claim language, the ’471 patent’s specification discloses “a method for processing a packet at an egress end user node” (EEUN) including a “decoding” step and a “concurrent writing” step “performed subsequent to the decoding step.” Ex. 1001, 3:6–11; *see id.* at code (57). The “decoding step involves decoding a packet having a plurality of headers.” *Id.* at 3:8–9; *see id.* at code (57). The “concurrent writing step involves concurrently writing (1) each of the headers to a packet buffer memory and (2) each individual one of the headers to a respective protocol

⁴ For Exhibit 2004 (Responsive Claim Construction Brief), we cite to the page numbers of the brief rather than the header page numbers.

stack layer memory where it is available for immediate processing within a protocol stack layer.” *Id.* at 3:11–15; *see id.* at code (57).

The ’471 patent’s specification also discloses an “egress end user node (EEUN)” including a decoder and a direct memory access (DMA) device coupled to the decoder. Ex. 1001, 3:45–47, 3:49. The “decoder is configured to decode a packet having a plurality of headers.” *Id.* at 3:47–49. The “DMA device is configured to concurrently write (1) each of the headers to a packet buffer memory and (2) each individual one of the headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.” *Id.* at 3:49–54.

Additionally, the ’471 patent’s specification explains that the DMA device “is used to concurrently write the specific layer header fields directly into: (a) each protocol stack layer’s internal memory spaces . . . and (b) the packet buffer memory.” Ex. 1001, 5:64–6:1; *see id.* at 6:38–43. More specifically, the DMA device “concurrently performs a plurality of write operations.” *Id.* at 9:45–47; *see id.* at 9:41–45. The concurrently performed “write operations” include:

- (1) “a write operation for concurrently communicating the media access control (MAC) layer protocol header 306 to the packet buffer memory 122 and the media access control (MAC) layer memory 126”;
- (2) “a write operation for concurrently communicating the logic link control (LLC) layer protocol header 308 to the packet buffer memory 122 and the logic link control (LLC) layer memory 128”;
- (3) “a write operation for concurrently communicating the network layer protocol header 310 to the packet buffer memory 122, the media access control (MAC) layer memory 126, and/or the network layer memory 130”;

- (4) “a write operation for concurrently communicating the transport layer protocol header 312 to the packet buffer memory 122 and the transport layer memory 132”; and
- (5) “a write operation for concurrently communicating the application layer header 314 to the packet buffer memory 122 and the application layer memory 134.”

Id. at 9:47–59, 9:65–10:4.

The ’471 patent’s specification similarly explains that the “DMA device concurrently communicates data to the different memories,” e.g., “the packet buffer memory 122, the media access control (MAC) layer memory 126, the logic link control (LLC) layer memory 128, the network layer memory 130, the transport layer memory 132, and the application layer memory 134” in Figure 1. Ex. 1001, 6:14–19, 6:22–24, Fig. 1; *see id.* at 6:38–43. The DMA device “allows the transfer of data to these various memory locations in a single DMA transaction.” *Id.* at 6:20–21.

For these reasons, the claim language and the ’471 patent’s specification support the construction of the “concurrently writing” requirement that we adopt, i.e., “concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.”

(b) Other Claim Terms, Phrases, and Limitations

“[O]nly those terms need be construed that are in controversy, and only to the extent necessary to resolve the controversy.” *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999); *see Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). We determine that no other claim term, phrase, or

limitation requires an express construction to decide whether Petitioner has shown by a preponderance of the evidence that claims 1–3, 5–10, 12–15, and 17–21 are unpatentable under § 103(a).

*D. Alleged Obviousness over Cornett, Paatela,
and Nelson: Claims 1–3, 5–10, 12–15, and 17–21*

Petitioner contends that claims 1–3, 5–10, 12–15, and 17–21 are unpatentable under § 103(a) because the claims would have been obvious over Cornett, Paatela, and Nelson. *See* Pet. 16–60; Reply 8–15. Patent Owner disputes Petitioner’s contentions. *See* Resp. 1–2, 18–36; Sur-reply 7–19.

Below, we provide overviews of Cornett, Paatela, and Nelson. Then, we consider the obviousness issues. For the reasons explained below, we agree with Petitioner that claims 1–3, 5–10, 12–15, and 17–21 are unpatentable under § 103(a) because the claims would have been obvious over Cornett, Paatela, and Nelson.

1. OVERVIEW OF CORNETT (EXHIBIT 1005)

Cornett is a U.S. patent application publication titled “Header Replication in Accelerated TCP (Transport Control Protocol) Stack Processing,” filed on May 26, 2005, and published on April 6, 2006. Ex. 1005, codes (12), (22), (43), (54). Cornett states that the invention relates to “accelerated TCP (Transport Control Protocol) stack processing.” *Id.* ¶ 3; *see id.* at code (57).

Cornett explains that “[e]ach memory access that occurs during packet processing may represent a potential delay as the processor awaits completion of the memory operation.” Ex. 1005 ¶ 5; *see id.* ¶ 63. For

instance, “using the core processing module of a host processor to perform slow memory access functions” introduces delay. *Id.* ¶ 5.

To accelerate packet processing and reduce delay, Cornett discloses “splitting” a packet into a header portion and a payload portion and then using a direct memory access (DMA) engine to separately store the header portion and the payload portion. Ex. 1005 ¶¶ 74, 97; *see id.* ¶¶ 34, 68, 73. Cornett also discloses a network interface controller (NIC) that may “include circuitry to perform header splitting.” *Id.* ¶ 79.

The NIC may also “include a physical layer device (PHY) that translates between the analog signals of a communications medium (e.g., a cable or wireless radio) and digital bits.” Ex. 1005 ¶ 79. “The PHY may be communicatively coupled to a media access controller (MAC) (e.g., via a FIFO) that performs ‘layer 2’ operations (e.g., Ethernet frame handling).” *Id.*

Cornett's Figure 7 (reproduced below) depicts steps in a process for storing packet information:

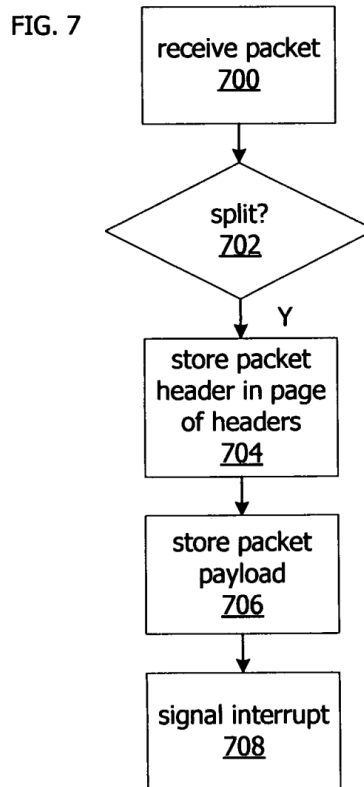


Figure 7 illustrates the operation of an NIC performing header splitting and separately storing a packet's header and payload. Ex. 1005 ¶ 73, Fig. 7; *see id.* ¶ 14.

According to Figure 7, after "receiving 700 a packet, the controller can determine 702 whether to perform header splitting." Ex. 1005 ¶ 73, Fig. 7. For example, "the controller may only perform splitting for TCP/IP packets or packets belonging to particular flows (e.g., particular TCP/IP connections or Asynchronous Transfer Mode (ATM) circuits)." *Id.* ¶ 73. "For packets selected for splitting, the controller can cause storage 704 (e.g., via Direct Memory Access (DMA)) of the packet's header in the page(s) used to store headers and separately store 706 the packet's payload." *Id.* ¶ 74, Fig. 7.

Cornett's Figure 9 (reproduced below) depicts a computer system:

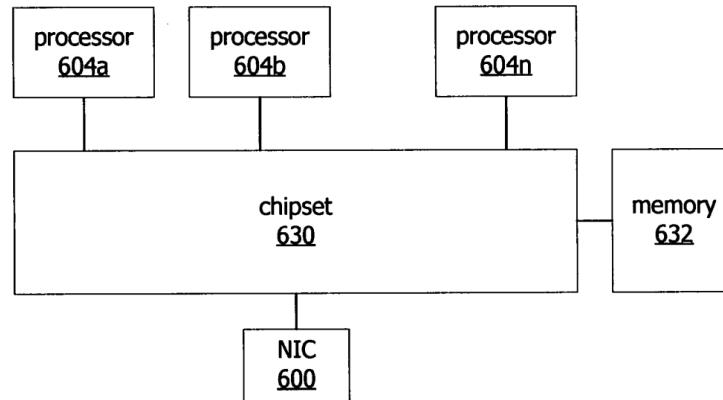


FIG. 9

Figure 9 illustrates “a sample computer architecture” of a system including “a chipset 630 that couples multiple processors 604a–604n to memory 632 and NIC 600.” Ex. 1005 ¶ 77, Fig. 9; *see id.* ¶ 16. “As shown, the system includes a single NIC 600.” *Id.* ¶ 79. But the “system may include multiple controllers,” and the “controller(s) can include a physical layer device (PHY).” *Id.*

Cornett's Figure 10 (reproduced below) depicts a computer system storing multiple packets:

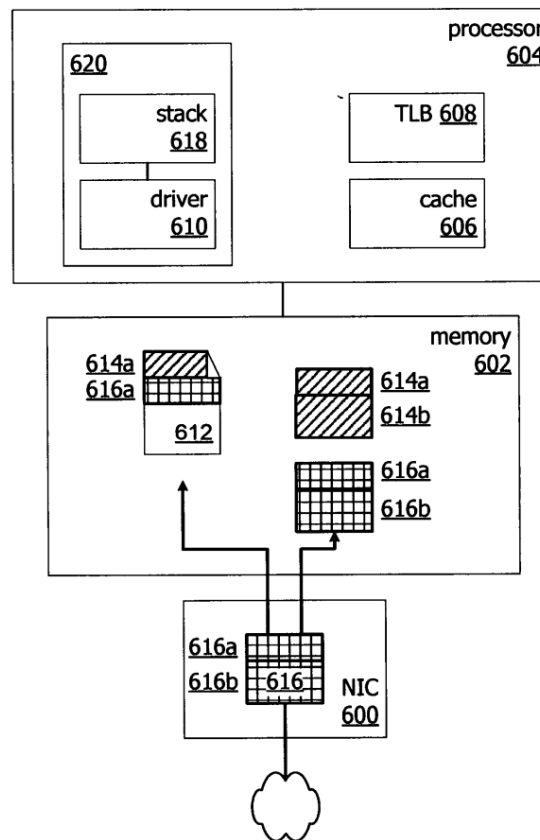


FIG. 10

Figure 10 illustrates a computer system including NIC 600, memory 602, and processor 604. *See* Ex. 1005 ¶¶ 17, 81, Fig. 10. Processor 604 includes cache 606, Translation Lookaside Buffer (TLB) 608, and driver 620, e.g., a “TCP driver that performs both accelerated and nonaccelerated processing.” *Id.* ¶¶ 66–67, Fig. 10.

As Figure 10 shows, NIC 600 (1) received an initial packet 614 having header 614a and payload 614b, (2) split packet 614 “into its constituent header 614a and payload 614b,” (3) wrote header 614a “into a physical page 612 of memory 602 used for storage of packet headers,” (4) wrote payload 614b “into a different location (e.g., a location not contiguous or in the same page as the location of the packet’s header 614a),”

and (5) stored header 614a at the location where payload 614b was written. *See* Ex. 1005 ¶¶ 68, 81, Fig. 10. As Figure 10 also shows, NIC 600 repeated that process for subsequently received packet 616 having header 616a and payload 616b. *See id.* ¶¶ 69, 81, Fig. 10.

Thus, Cornett discloses that “the payload 614b, 616b may be stored to a first location, while the header 614a, 616a may be stored to the first location, as well as a second location different from the first location.” Ex. 1005 ¶ 81, Fig. 10.

Additionally, Cornett discloses that “TCP/IP processing may be accelerated by using a data movement module, such as a DMA engine, to move data from one buffer to another buffer.” Ex. 1005 ¶ 97. “Since the core processing module of a host processor may be bypassed using a DMA engine, slow memory access speeds may be avoided.” *Id.* A host processor may include a DMA engine as a “support module.” *Id.* ¶ 49.

2. OVERVIEW OF PAATELA (EXHIBIT 1007)

Paatela is a U.S. patent application publication titled “System and Method for Providing Transformation of Multi-Protocol Packets in a Data Stream,” filed on September 12, 2005, and published on September 21, 2006. Ex. 1007, codes (12), (22), (43), (54). Paatela states that the invention “relates in general to communication networks, and more particularly to a method and apparatus for transforming packets in a multi-protocol, multi-flow data stream.” *Id.* ¶ 5; *see id.* at code (57).

Paatela explains that data transmitted over networks “is generally broken up into a number of data packets, each of which is assigned a hierarchy of headers to direct the data packet to the desired destination, among other things.” Ex. 1007 ¶ 39. Packets “transmitted across a network

are associated with a transmission protocol,” i.e., “a set of rules that governs how devices on a network exchange information.” *Id.* ¶ 8.

Paatela also explains that because “routers take on the task of intercepting, analyzing, and moving on millions of packets per second along the best possible route, the processing occurring at these routers must be extremely efficient to avoid bogging down the system.” Ex. 1007 ¶ 41. According to Paatela, the invention “may be used in connection with such routing systems, increasing speed and efficiencies of network data throughput.” *Id.*

More specifically, packets communicated across a network using different transmission protocols “can be appropriately edited regardless of the transmission protocol associated with the packets.” Ex. 1007 ¶ 38; *see id.* ¶¶ 15, 17. “Portions of each packet that are subject to change (but may not necessarily be changed) are temporarily stored.” *Id.* ¶ 38; *see id.* at code (57). “Certain instructions for effecting appropriate modifications to the particular packet are processed, with due consideration to the packet’s protocol, which results in a protocol-dependent modification of the temporarily stored packet information.” *Id.* ¶ 38; *see id.* at code (57). “Validity tags are associated with different segments of the temporarily-stored packet, where the state of each tag determines whether its corresponding packet segment will form part of the resulting modified packet.” *Id.* ¶ 38; *see id.* at code (57). “Those packet segments identified as being part of the resulting modified packet are reassembled prior to dispatch of the packet.” *Id.* ¶ 38; *see id.* at code (57).

Paatela's Figure 7 (reproduced below) depicts an editing system for packets communicated across a network using different transmission protocols:

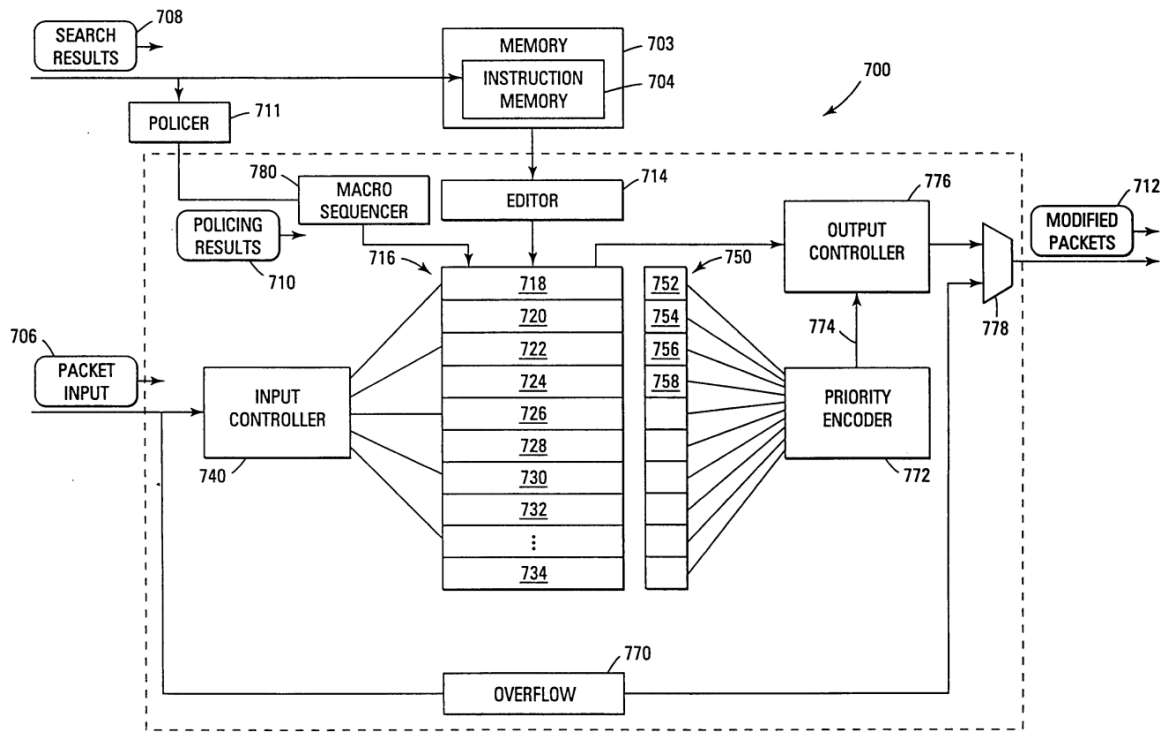


Fig. 7

Figure 7 illustrates “an editing system 700” (“also referred to as a packet transformation system”) including editing module 702 (not numbered but identified by a dashed rectangle), memory 703, and policer 711. Ex. 1007 ¶¶ 59–60, Fig. 7; *see id.* ¶ 26. Editing module 702 includes editing processor 714, memory 716 comprising a plurality of memory locations 718–734, input controller 740, valid bit array 750, overflow buffer 770, priority encoder 772, output controller 776, multiplexer 778, and macro sequencer 780. *Id.* ¶¶ 76–77, 85, 88–89, 91, Fig. 7. Memory 703 includes instruction memory 704. *Id.* ¶¶ 60–61, Fig. 7.

“Inputs to the editing system 700 include packet/frame input ultimately originating from the pre-processor or the classifier and labeled as

the ‘packet input’ 706.” Ex. 1007 ¶ 60, Fig. 7. “The editing system 700 outputs modified packets 712,” e.g., “to an elastic queue.” *Id.* ¶ 60.

The “instructions/data from the memory 703 are processed by the editing processor 714, and operate on data stored in the memory 716.” Ex. 1007 ¶ 76. Memory 716 “includes a plurality of memory locations,” illustrated in Figure 7 as “memory locations 718, 720, 722, 724, 726, 728, 730, and 732 through some finite number of memory locations up to the end memory location 734.” *Id.* ¶ 76, Fig. 7.

Packet 706 “is stored in the memory 716 in time for the editing instructions from the instruction memory 704 to be processed by the editing processor 714, and for the packet data in the memory 716 to be operated on by the editing processor 714.” Ex. 1007 ¶ 76.

Memory locations 718–734 are “partitioned such that at least some of these segments are allocated to store data corresponding to certain portions of the packet(s) 706,” such as portions corresponding to “headers of the various protocol layers associated with the incoming packet.” Ex. 1007 ¶ 77. For instance, “header information corresponding to OSI networking layers two through four may each correspond to a segment of the memory 716,” such that:

a segment of memory 716 is allocated to store a layer-2 header (e.g., a PPP header), a segment corresponding to a layer-2.5 header (e.g., an MPLS label stack), a segment corresponding to a layer-3 header (e.g., an IP header), and a segment corresponding to a layer-4 header (e.g., a TCP header).⁵

⁵ The acronym “PPP” stands for “Point-to-Point Protocol.” *See, e.g.*, Ex. 1007 ¶¶ 125, 130; Ex. 1009, 249. The acronym “MPLS” stands for “Multi-Protocol Label Switching.” *See, e.g.*, Ex. 1033, 1; Ex. 1035 ¶ 52.

Id.

To allow “selected ones of the allocated memory segments to be edited,” allocated memory locations 718–734 are interleaved with unused memory locations 718–734, i.e., memory locations “unused during the input stage.” Ex. 1007 ¶ 79; *see id.* ¶¶ 113, 120, 125, 130, Figs. 13–16. For instance, “if memory location 718 stores an Ethernet header and memory location 722 stores an IPv4 header, the Ethernet header can be modified by writing a new Ethernet header into an otherwise unused, interleaved memory location 720, and disregarding the original Ethernet header in memory location 718.” *Id.* ¶ 79. “When the memory locations are read out in the proper order, the new Ethernet header in memory location 720 effectively replaces the original (now-disregarded) Ethernet header at memory location 718.” *Id.*

In “one embodiment,” memory 716 “is a dual-port memory.” Ex. 1007 ¶ 78. “A dual-port memory can be simultaneously read and/or written by two different data sources, or more generally, a shared memory accessible by two processes.” *Id.*

In “one embodiment,” the “data stream is a 64-bit data stream, and the memory 716 is a 32-bit wide, 128-word deep, dual-port memory.” Ex. 1007 ¶ 78. “In this manner, two 32-bit words may be simultaneously written to the memory 716 to write the 64-bit data.” *Id.* “Alternatively, data can be concurrently written to and read from the memory 716,” e.g., thus

The acronym “IP” stands for “Internet Protocol.” *See, e.g.*, Ex. 1001, 1:16–18; Ex. 1003 ¶ 27. The acronym “TCP” stands for “Transmission Control Protocol.” *See, e.g.*, Ex. 1001, 1:17–18; Ex. 1003 ¶ 27; Ex. 1008, 5.

“‘overlapping’ the input write stage of the processing with the output read stage of the processing.” *Id.*

Paatela discloses using a quad-port memory or multi-port memory as an alternative to a dual-port memory to “produce analogous results and provide similar advantages” as a dual-port memory. Ex. 1007 ¶ 78.

3. OVERVIEW OF NELSON (EXHIBIT 1006)

Nelson is a U.S. patent titled “System and Method for Storing Frame Header Data,” filed on October 24, 2000, and issued on February 19, 2008. Ex. 1006, codes (12), (22), (45), (54). Nelson states that the invention relates to “a system and method implemented in connection with packet switching protocols wherein packet (frame) header information is stored in a buffer separate from the corresponding packet buffer to allow routing decisions to be made based on the header information without reading the packet out of the buffer.” *Id.* at 1:13–20; *see id.* at 2:3–10, code (57).

Nelson explains that when “a frame is received by a particular switch, the switch typically stores the frame (including the corresponding header) in a buffer until it can be sent to another device.” Ex. 1006, 1:46–48. “This buffer is normally a first-in-first-out (FIFO) buffer.” *Id.* at 1:51–52. “When a frame reaches the head of the FIFO buffer, it can be routed (transmitted) by the switch.” *Id.* at 1:52–53.

Nelson also explains that “the latency of the frame’s transport from the source device to the destination device is increased” because (1) “the transport information is stored only in the header of the frame which is in the FIFO,” (2) “at least part of the frame must be read out of the FIFO so that the transport information can be read,” and (3) the information read out of

the FIFO must be stored in a second location “while the routing decision is made.” Ex. 1006, 1:57–67.

Nelson endeavors to reduce “the latency of the frame’s transport” by providing “a system and method for storing header information in parallel with corresponding packets or frames of data” so that header information may be accessed separately from packet or frame storage to make a routing decision. *See* Ex. 1006, 2:3–21, 3:16–22, 3:40–44, 5:39–42, 5:50–56. For example, overlapping “the routing decision for a frame with the transmission of the preceding frame” eliminates delay, “thereby reducing the latency in transporting the frames through the port.” *Id.* at 5:39–42; *see id.* at 5:50–56. Additionally, “the throughput of the system may be increased” because “a routing decision can be made before the frame is ready to be transmitted.” *Id.* at 3:46–48.

In “one embodiment,” the “system is implemented in a Fibre Channel switch” where each “Fibre Channel link consists of a pair of unidirectional fibers” and each “fiber is connected to an associated transmitter and receiver.” Ex. 1006, 3:49–50, 4:4–6. Nelson’s Figure 2 (reproduced below) depicts a Fibre Channel fabric formed from multiple interconnected Fibre Channel links and switches:

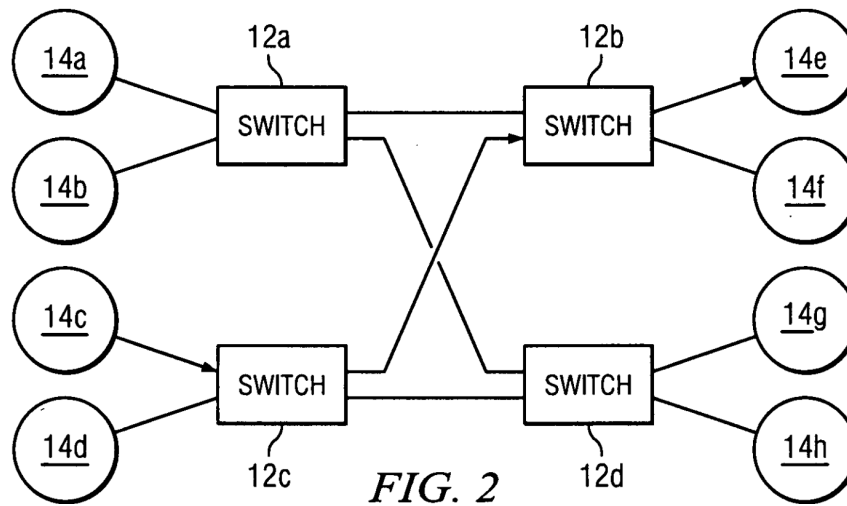


Figure 2 illustrates a Fibre Channel fabric including switches 12a–12d and sites 14a–14h configured to (1) “receive and buffer frames of data from a source,” (2) “make routing decisions for the frames,” and (3) “transmit the frames to the appropriate destination.” Ex. 1006, 4:14–21, Fig. 2. In Figure 2, “frames are shown being routed from site 14c, through switch 12c, then through switch 12b, and finally to destination site 14e.” *Id.* at 4:21–23, Fig. 2.

Nelson’s Figure 3 (reproduced below) depicts a frame structure:

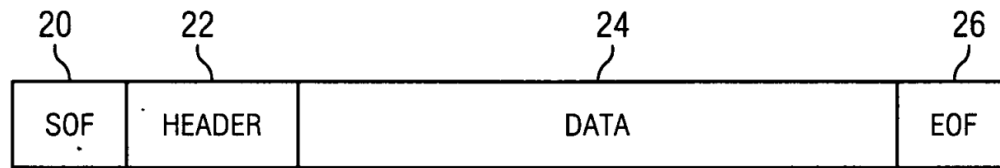


FIG. 3

Figure 3 illustrates a frame in “one embodiment” that includes “a start-of-frame (SOF) primitive 20, a header 22, data 24, and an end-of-frame (EOF) primitive 26.” Ex. 1006, 2:51–52, 4:34–38, Fig. 3. SOF primitive 20 “identifies the beginning of a frame.” *Id.* at 4:38–39. EOF primitive 26 “similarly serves to identify the end of the frame.” *Id.* at 4:39–40.

Header 22 “comprises overhead data related to the frame.” Ex. 1006, 4:42–43. Data 24 “comprises useful data.” *Id.* at 4:47. “All of the data of a frame, whether useful data or overhead data, is enclosed between the SOF and EOF primitives.” *Id.* at 4:40–42.

Nelson’s Figure 5 (reproduced below) depicts a system for storing header information in parallel with corresponding packets or frames of data:

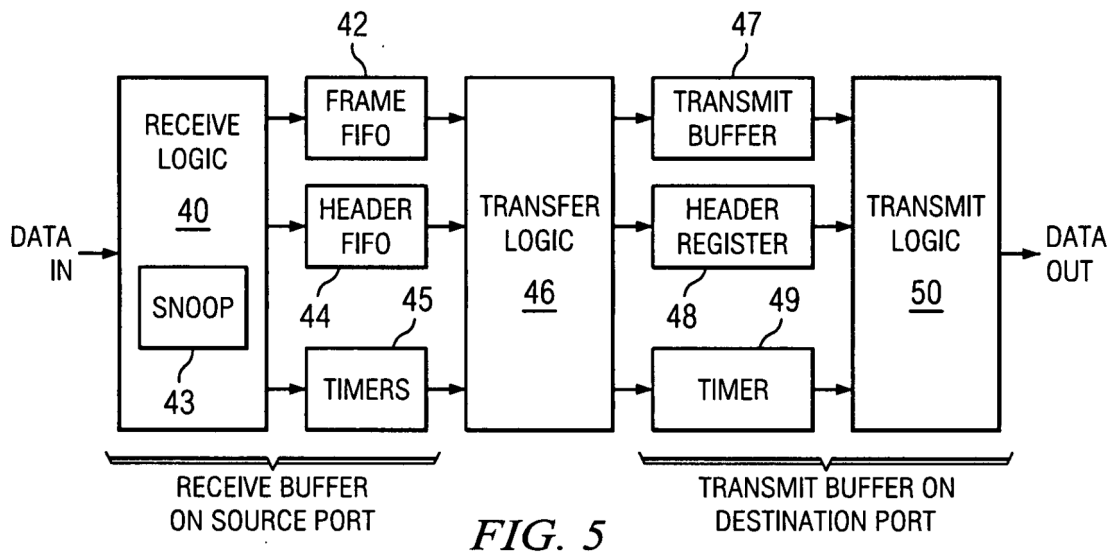


Figure 5 illustrates “an embodiment” of Nelson’s system that includes the following components:

- receive logic 40 “configured to detect the SOF and EOF primitives which define the boundaries of frames in the received data and to store individual frames”;
- frame FIFO 42;
- header FIFO 44;
- timers 45;
- transfer logic 46 “configured to make routing decisions for the frames that are received by a port”;
- transmit buffer 47;
- header register 48;

- timer 49; and
- transmit logic 50 that transmits the frame in transmit buffer 47 and examines the header in header register 48 “in preparation for transmitting” the next frame.

Ex. 1006, 6:59–66, 7:8–39, Fig. 5.

Receive logic 40 “includes circuitry to snoop on the received data and to copy the header of each frame to header FIFO 44.” Ex. 1006, 6:66–7:1; *see id.* at 7:52–57, Fig. 6. “The copying of a header to header FIFO 44 occurs in parallel with the storage of the corresponding frame in frame FIFO 42.” *Id.* at 7:1–4. Hence, “the header information is stored” at “the same time the frame is stored.” *Id.* at 6:29–30; *see id.* at 7:52–57, Fig. 6.

Nelson explains that in “another embodiment,” a dual-port random-access memory (RAM) “can be used to store the header information” instead of a FIFO. Ex. 1006, 6:8–9; *see id.* at 6:44–45. “While RAM implementations do not have the same physical first-in-first-out constraints of FIFO memory devices,” the same first-in-first-out behavior may be achieved “using pointer manipulation.” *Id.* at 6:9–12, 6:20–21. “Headers in the RAM are accessed in a circular manner with read and write pointers.” *Id.* at 6:12–13. Further, “the frame FIFO can also be implemented using these different types of memory elements.” *Id.* at 6:16–17.

4. WHETHER AN ORDINARILY SKILLED ARTISAN WOULD
HAVE BEEN MOTIVATED TO COMBINE THE TEACHINGS OF
THE REFERENCES WITH A REASONABLE EXPECTATION OF SUCCESS

(a) Motivation to Combine

Petitioner contends that an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, and Nelson because “they all aim to accelerate packet processing.” Pet. 24 (citing Ex. 1003 ¶ 56;

Ex. 1005 ¶ 97; Ex. 1006, 1:13–20, 2:3–10; Ex. 1007 ¶¶ 76, 79, code (57)); *see id.* at 19, 21, 23 (citing Ex. 1001, 1:8–11; Ex. 1003 ¶¶ 48, 52, 55; Ex. 1005 ¶ 97; Ex. 1006 3:20–22; Ex. 1007 ¶ 41). According to Petitioner, “[t]here is motivation to combine where a known technique ‘has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way.’” *Id.* at 24 (quoting *Intel Corp. v. Qualcomm Inc.*, 21 F.4th 784, 800 (Fed. Cir. 2021)).

More specifically, Petitioner asserts that “Cornett teaches why accelerated packet processing is beneficial to a network and discloses a technique for doing so by storing packet headers in one buffer and storing the entire packet (header and payload) in another, separate buffer.” Pet. 25 (citing Ex. 1003 ¶ 58). According to Petitioner, storing packet headers in one buffer and storing the entire packet (header and payload) in another, separate buffer:

- (1) “reduces the number of memory pages ‘in a way to permit effective prefetching of packet headers into the processor cache before the protocol stack processes the header’”; and
- (2) “significantly reduce[s] TCP/IP processing overhead that may result from using the core processing module of a host processor.”

Id. at 27, 29 (alteration by Petitioner) (quoting Ex. 1005 ¶¶ 64, 97).

Petitioner concedes that “Cornett does not explicitly disclose how each header of a packet is stored or how each such header is processed.” Pet. 25 (citing Ex. 1003 ¶ 58); *see id.* at 38 (citing Ex. 1003 ¶ 83). But Petitioner contends that “Paatela provides the missing detail” because “Paatela teaches separating out each of a packet’s headers and writing each such header to a respective, separate memory segment for processing.” *Id.*

at 25–27 (citing Ex. 1003 ¶ 58; Ex. 1007 ¶ 77); *see id.* at 40–41 (citing Ex. 1003 ¶ 86).

Petitioner explains that Paatela’s memory 716 includes multiple segments or memory locations 718–734 that may store “headers of the various protocol layers associated with the incoming packet.” Pet. 27 (quoting Ex. 1007 ¶ 77); *see id.* at 39–40. Petitioner also explains that in the Cornett-Paatela combination “Cornett’s NIC would cause each individual packet header to be stored into a respective, separate segment of memory for processing, per Paatela.” *Id.* at 41 (citing Ex. 1003 ¶ 86; Ex. 1007 ¶ 77).

Petitioner admits that the “Cornett-Paatela combination does not explicitly explain the timing of how packet headers are written to Paatela’s respective, separate memory segments.” Pet. 26 (citing Ex. 1003 ¶ 58); *see id.* at 29, 42 (citing Ex. 1003 ¶ 64). But Petitioner asserts that an ordinarily skilled artisan would have “looked to Nelson to provide the missing detail that is consistent with the Cornett-Paatela combination.” *Id.* at 30 (citing Ex. 1003 ¶ 65). Petitioner asserts that an ordinarily skilled artisan “would have appreciated that Nelson presents an obvious and highly logical design choice consistent with the Cornett-Paatela combination” for “how to further the Cornett-Paatela combination’s accelerated packet processing technique.” *Id.* at 30–31 (citing Ex. 1003 ¶¶ 65–66).

Regarding the Cornett-Paatela-Nelson combination, Petitioner explains that:

- (1) Cornett seeks to reduce “processing overhead and processing time”;
- (2) Paatela focuses on “optimizing packet processing in an environment where time is of the essence”;

- (3) “Nelson seeks to accelerate packet processing and reduce processing overhead” by “writing a packet frame (header and payload) into one buffer and the packet headers into another buffer, with the two writing operations occurring in parallel”; and
- (4) “Nelson’s parallel writing would further accelerate packet processing, which is Cornett’s and Paatela’s goal as well.”

Pet. 30, 43–44 (citing Ex. 1003 ¶¶ 65, 90; Ex. 1005 ¶ 97; Ex. 1006, 7:1–4; Ex. 1007 ¶ 76); *see* Reply 13.

Further, Petitioner asserts that the Cornett-Paatela-Nelson combination “does not require physical incorporation of elements from Paatela and Nelson into Cornett.” Pet. 31 (citing Ex. 1003 ¶ 67). Although the Cornett-Paatela-Nelson combination does not require physical incorporation, Petitioner asserts that physical incorporation would have been possible and would have yielded the claimed invention. *Id.* at 31–32 (citing Ex. 1003 ¶ 67); *see id.* at 28 (citing Ex. 1003 ¶ 63); Reply 13–14. Petitioner explains that physical incorporation would use:

- (1) “Paatela’s input controller and packet classification engine to ‘determine[] where one networking layer header ends and the next networking layer header begins’”; and
- (2) “Cornett’s DMA engine to move each individual packet header into a respective memory segment, per Paatela”; and
- (3) a “parallel” move operation for the packet headers per Nelson “to further accelerate packet processing.”

Pet. 28, 31–32 (alteration by Petitioner) (quoting Ex. 1007 ¶ 77) (citing Ex. 1003 ¶ 67; Ex. 1005 ¶¶ 74, 97; Ex. 1006, 7:1–4); Reply 14–15 (quoting

Ex. 1003 ¶¶ 67, 135); *see* Pet. 29, 44 (citing Ex. 1003 ¶¶ 64, 91; Ex. 1005 ¶ 81; Ex. 1006, 6:66–7:4; Ex. 1007 ¶ 77).

According to Petitioner, “[w]riting each of the protocol headers to its respective protocol header memory in parallel would have been obvious because doing so would ‘accelerate packet processing.’” Reply 15 (citing Ex. 1003 ¶¶ 58, 65–66; Ex. 1006, 7:1–4).

Additionally, Petitioner contends that the Cornett-Paatela-Nelson combination “represents the use of known techniques,” specifically:

- (1) “writing packet headers into individual memory segments to accelerate processing, per Paatela”; and
- (2) “writing an entire packet frame into one buffer and the packet headers only into another buffer in parallel to accelerate routing decisions, per Nelson.”

Pet. 25 (citing Ex. 1003 ¶ 57). Petitioner contends that using these known techniques would “improve a similar process (accelerated packet processing technique that stores packet headers separately from an entire packet—both header and payload—using a DMA engine, per Cornett) in the same way to yield predictable results.” *Id.* (citing Ex. 1003 ¶ 57).

Patent Owner makes no arguments against combining the teachings of Cornett, Paatela, and Nelson. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, and Nelson in the way Petitioner proposes. *See* Pet. 24–32; Ex. 1003 ¶¶ 56–61, 63–65, 67.

(b) Reasonable Expectation of Success

Petitioner asserts that an ordinarily skilled artisan would have had a reasonable expectation of success in combining Paatela's teachings with Cornett's teachings because (1) "Cornett teaches how to process packet headers separately from the entire packet (header and payload)" and (2) "Paatela merely adds detail about how each header is individually stored in a respective, separate memory segment" that Cornett expects an ordinarily skilled artisan to already know. Pet. 28–29 (citing Ex. 1003 ¶ 62; Ex. 1005 ¶¶ 64, 81; Ex. 1007 ¶ 77). According to Petitioner, "Cornett's use of a DMA engine to move data from one buffer to another would simply offload data movement responsibilities from the core processing module of a host processor." *Id.* at 29 (citing Ex. 1005 ¶ 97).

Petitioner asserts that an ordinarily skilled artisan would have had a reasonable expectation of success in combining Nelson's teachings with the teachings of Cornett and Paatela because the skilled artisan would have appreciated that:

- (1) "Cornett seeks to further accelerate packet processing by using a DMA engine to free up the host processor and move data more efficiently";
- (2) "Paatela stores individual packet headers in respective, separate memory segments to make each packet header available for processing as quickly as possible";
- (3) Nelson, like Cornett, "seeks to process packets faster by writing headers and frames into separate buffers—in parallel—so that header information is made available for routing decisions without the need (and associated delay) to read the entire frame from the buffer"; and

- (4) Nelson discloses “a logical design choice for how to further the Cornett-Paatela combination’s accelerated packet processing technique.”

Pet. 31 (citing Ex. 1003 ¶ 66; Ex. 1005 ¶ 97; Ex. 1006, 2:3–10; Ex. 1007 ¶ 76).

Petitioner also asserts that the Cornett-Paatela-Nelson combination “would have been within the skillset” of an ordinarily skilled artisan because the combination rests on “the teachings of the references themselves.” Pet. 29, 31 (citing Ex. 1003 ¶¶ 62, 67).

Patent Owner does not dispute that an ordinarily skilled artisan would have had a reasonable expectation of success in combining the teachings of Cornett, Paatela, and Nelson. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that an ordinarily skilled artisan would have had a reasonable expectation of success in combining the teachings of Cornett, Paatela, and Nelson. *See* Pet. 28–29, 31–32; Ex. 1003 ¶¶ 57, 62–63, 66–67.

5. INDEPENDENT CLAIM 13

(a) Preamble 13.0

Claim 13’s preamble recites “[a]n egress end user node (EEUN) of a packet based communications system.” Ex. 1001, 12:33–34.

Petitioner contends that Cornett teaches claim 13’s preamble because Cornett discloses “a network consisting of a ‘plurality of nodes’ that can be ‘one or more intermediate stations, such as, for example, one or more hubs, switches, and/or routers . . . [and] one or more end stations.’” Pet. 32 (alterations by Petitioner) (quoting Ex. 1005 ¶¶ 21, 23) (citing Ex. 1005 ¶ 4). Petitioner contends that each network node in Cornett includes a network

interface controller (NIC). *Id.* (citing Ex. 1005 ¶¶ 28, 77). Therefore, according to Petitioner, Cornett’s network includes “[a]n egress end user node (EEUN)” as recited in claim 13’s preamble. *Id.*

As support, Petitioner provides the following annotated version of Cornett’s Figure 9 (Pet. 33):

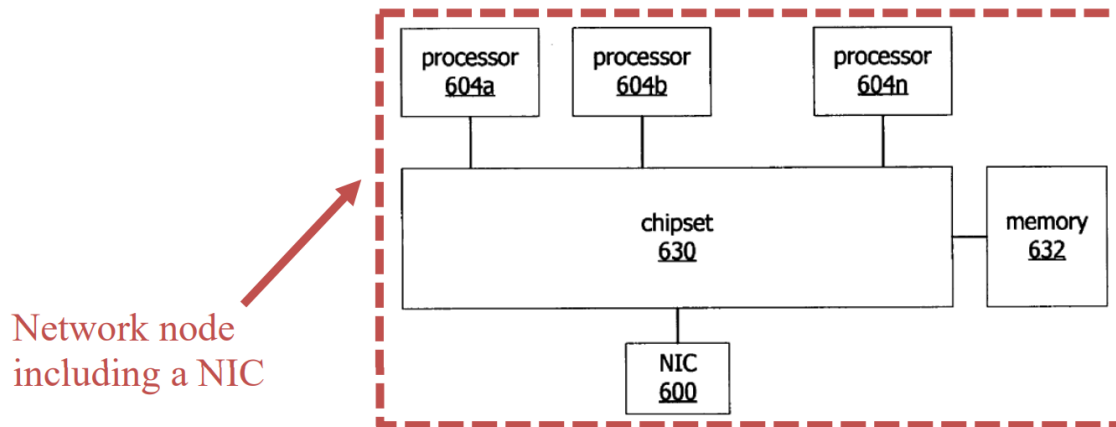


Figure 9 illustrates “a sample computer architecture” of a system including “a chipset 630 that couples multiple processors 604a–604n to memory 632 and NIC 600.” Ex. 1005 ¶ 77, Fig. 9; *see id.* ¶ 16. The above annotated version of Figure 9 includes:

- (1) a dashed red rectangle around the system including NIC 600, processors 604a–604n, chipset 630, and memory 632; and
- (2) the annotation “Network node including a NIC” with a red arrow identifying the system including NIC 600, processors 604a–604n, chipset 630, and memory 632.

See Pet. 33; Ex. 1003 ¶ 70.

Petitioner also contends that “Cornett’s nodes communicate ‘via wirelined (e.g., copper wires), or wireless (e.g., radio frequency) means’ using ‘one or more packets.’” Pet. 33 (quoting Ex. 1005 ¶ 21). Therefore, according to Petitioner, Cornett’s network corresponds to “a packet based

communications system” as recited in claim 13’s preamble. *Id.* (citing Ex. 1003 ¶ 69).

Patent Owner makes no arguments specific to claim 13’s preamble. *See, e.g.*, Resp. 18–38; Sur-reply 7–21. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

Generally, a preamble does not limit a claim. *Allen Eng’g Corp. v. Bartell Indus., Inc.*, 299 F.3d 1336, 1346 (Fed. Cir. 2002). We need not decide whether claim 13’s preamble limits the claim because, for the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches claim 13’s preamble. *See* Pet. 32–33; Ex. 1003 ¶¶ 68–72.

(b) Limitation 13.1

Claim 13 requires “a decoder configured for decoding a packet having a plurality of headers.” Ex. 1001, 12:35–36 (limitation 13.1).

Petitioner contends that Cornett teaches limitation 13.1 because Cornett’s NIC includes “a physical layer device (PHY)” that “translates between the analog signals of a communications medium (e.g., a cable or wireless radio) and digital bits” to make “a packet readable by digital processing logic in Cornett’s ‘nodes,’ which are devices that process digital bits.” Pet. 33–34 (quoting Ex. 1005 ¶ 79) (citing Ex. 1003 ¶¶ 73–74). Petitioner explains that the PHY (1) converts “analog-digital representations of a packet as it is transmitted on the communication medium into digital bits” and (2) corresponds to “a decoder configured for decoding a packet” according to limitation 13.1. *Id.* (citing Ex. 1003 ¶ 73); *see id.* at 35 (citing Ex. 1003 ¶ 77; Ex. 1005 ¶ 79).

Further, Petitioner asserts that an ordinarily skilled artisan would have:

- (1) “known that packets typically include a plurality of headers because a protocol operating on a particular network communication layer adds its own header to the packet, encapsulating the headers from the lower layers”; and
- (2) “understood that Cornett’s packets” include “a plurality of headers” according to limitation 13.1.

Pet. 34 (citing Ex. 1003 ¶ 75; Ex. 1005 ¶ 3; Ex. 1009, 19–20; Ex. 1023, 4).

Patent Owner makes no arguments specific to limitation 13.1. *See, e.g.,* Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches limitation 13.1. *See* Pet. 33–34; Ex. 1003 ¶¶ 73–75.

(c) Limitation 13.2

Claim 13 requires “a direct memory access (DMA) device coupled to said decoder.” Ex. 1001, 12:37–38 (limitation 13.2).

Petitioner contends that Cornett teaches limitation 13.2 because Cornett’s NIC performs “header splitting” on selected packets where “the NIC ‘can cause storage 704 (e.g., via Direct Memory Access (DMA)) of the packet’s header.’” Pet. 34–35 (quoting Ex. 1005 ¶ 74) (citing Ex. 1003 ¶ 76; Ex. 1005 ¶¶ 73, 97). Petitioner also contends that Cornett’s DMA engine corresponds to “a direct memory access (DMA) device” according to limitation 13.2. *Id.* at 35 (citing Ex. 1003 ¶ 76).

Also, Petitioner explains that Cornett’s NIC includes a PHY that (1) converts “analog-digital representations of a packet as it is transmitted on the communication medium into digital bits” and (2) corresponds to “a

decoder configured for decoding a packet” according to limitation 13.1. Pet. 35 (citing Ex. 1003 ¶ 77; Ex. 1005 ¶ 79); *see id.* at 33–34 (citing Ex. 1003 ¶ 73). Petitioner contends that Cornett’s NIC converts the packet headers into digital bits and then “causes storage of the digital bits of the packet headers into memory using the DMA engine.” *Id.* at 35 (citing Ex. 1003 ¶ 77; Ex. 1005 ¶ 74).

Further, Petitioner asserts that an ordinarily skilled artisan would have found it obvious that Cornett’s DMA engine is “coupled to” Cornett’s PHY as required by limitation 13.2. Pet. 35 (emphasis omitted) (citing Ex. 1003 ¶ 77).

Patent Owner makes no arguments specific to limitation 13.2. *See, e.g.,* Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches limitation 13.2. *See* Pet. 34–35; Ex. 1003 ¶¶ 76–77.

(d) Limitation 13.3

Claim 1 requires a DMA device “configured for concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.” Ex. 1001, 12:37–43 (limitation 13.3).

Petitioner contends that the combined disclosures in Cornett, Paatela, and Nelson teach limitation 13.3 for several related reasons. *See* Pet. 35–45; Reply 8–15.

(i) Writing Each of the Headers to a Packet Buffer Memory

Initially, regarding the requirement for a DMA device configured for “writing . . . each of said plurality of headers to a packet buffer memory,” Petitioner asserts that Cornett’s NIC:

- (1) receives packet 616 having header 616a and payload 616b;
- (2) stores header 616a in a buffer along with previously received and stored header 614a;
- (3) stores payload 616b in a buffer along with previously received and stored payload 614b separate from the buffer storing headers 614a and 616a; and
- (4) also stores headers 614a and 616a to the buffer storing payloads 614b and 616b such that the buffer contains “the entire packet (header and payload, 614a and b and 616a and b).”

Pet. 36–37 (citing Ex. 1003 ¶¶ 79–80; Ex. 1005 ¶¶ 68, 81, Fig. 10).

Petitioner explains that “[t]he buffer where the entire packet (header and payload) is stored is the claimed packet buffer memory because a plurality of headers is stored there.” *Id.* at 37 (emphases omitted) (citing Ex. 1003 ¶ 80).

As support, Petitioner provides the following highlighted and annotated version of Cornett's Figure 10 (Pet. 37):

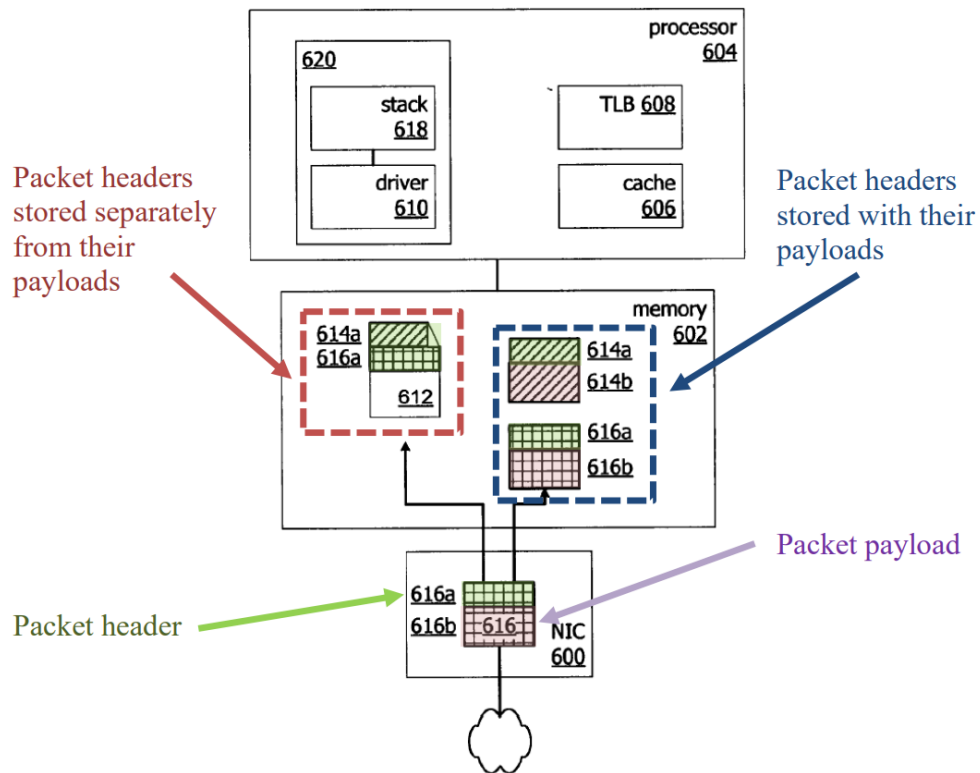


Figure 10 illustrates a computer system including NIC 600, memory 602, and processor 604 where NIC 600 has received initial packet 614 having header 614a and payload 614b and subsequent packet 616 having header 616a and payload 616b. *See* Ex. 1005 ¶¶ 17, 68–69, 81, Fig. 10. The above highlighted and annotated version of Figure 10 includes:

- (1) green highlighting over header 614a in memory 602;
- (2) green highlighting over header 616a in NIC 600 and memory 602;
- (3) pink highlighting over payload 614b in memory 602;
- (4) pink highlighting over payload 616b in NIC 600 and memory 602;
- (5) the green annotation “Packet header” with a green arrow identifying header 616a in NIC 600;

- (6) the purple annotation “Packet payload” with a purple arrow identifying payload 616b in NIC 600;
- (7) a dashed red rectangle around the locations in memory 602 storing headers 614a and 616a;
- (8) the red annotation “Packet headers stored separately from their payloads” with a red arrow identifying the dashed red rectangle around the locations in memory 602 storing headers 614a and 616a;
- (9) a dashed blue rectangle around the locations in memory 602 storing header 614a along with payload 614b and header 616a along with payload 616b; and
- (10) the blue annotation “Packet headers stored with their payloads” with a blue arrow identifying the dashed blue rectangle around the locations in memory 602 storing header 614a along with payload 614b and header 616a along with payload 616b.

See Pet. 37; Ex. 1003 ¶ 79.

Further, Petitioner asserts that an ordinarily skilled artisan “would have understood, or at least it would have been obvious,” that Cornett’s DMA engine writes “packet headers and the entire packets (header and payload) to memory” because “Cornett explains that its invention may significantly reduce TCP/IP processing overhead by using ‘a data movement module, such as a DMA engine, to move data from one buffer to another buffer.’” Pet. 37–38 (quoting Ex. 1005 ¶ 97) (citing Ex. 1003 ¶ 81). According to Petitioner, this explanation would express to an ordinarily skilled artisan that “Cornett expects to use the DMA engine throughout its invention.” *Id.* at 38 (citing Ex. 1003 ¶ 81).

Petitioner also asserts that Cornett’s Figure 7 embodiment “explicitly teaches using the DMA engine to write headers and payloads to memory.” Pet. 38 (citing Ex. 1003 ¶ 81).

Therefore, according to Petitioner, Cornett teaches a DMA engine configured for “writing . . . each of said plurality of headers to a packet buffer memory” according to limitation 13.3. Pet. 38 (citing Ex. 1003 ¶ 81).

Patent Owner does not argue that Cornett fails to teach a DMA engine configured for “writing . . . each of said plurality of headers to a packet buffer memory” according to limitation 13.3. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches a DMA engine configured for “writing . . . each of said plurality of headers to a packet buffer memory” according to limitation 13.3. *See* Pet. 35–38; Ex. 1003 ¶¶ 78–82.

(ii) Writing Each of the Individual Headers to a Respective Protocol Stack Layer Memory

Next, regarding the requirement for “writing . . . each individual one of said plurality of headers to a respective protocol stack layer memory,” Petitioner asserts that Paatela “teaches routing packets using a process called ‘packet transformation’ where packets are ‘temporarily stored, and acted upon through processing of protocol-dependent instructions.’” Pet. 38 (quoting Ex. 1007, code (57)).

Further, Petitioner asserts that Paatela discloses:

- (1) system 700 receiving packets/frames 706 as inputs;
- (2) input controller 740 determining “where one networking layer header ends and the next networking layer header

begins” and then using “this information to direct header information to memory 716”;

- (3) memory 716 “configured and partitioned such that all header information and data is stored within” memory 716; and
- (4) memory 716 storing “portions of the packet input 706 . . . in memory locations 718–734” that correspond to “headers of the various protocol layers associated with the incoming packet.”

Pet. 38–39 (alteration by Petitioner) (quoting Ex. 1007 ¶¶ 76–77, 85).

Additionally, Petitioner quotes Paatela’s disclosure about memory locations 718–734 storing header information as follows:

header information corresponding to OSI networking layers two through four may each correspond to a segment of the memory 716, such that a segment of memory 716 is allocated to store a layer-2 header (e.g., a PPP header), a segment corresponding to a layer-2.5 header (e.g., an MPLS label stack), a segment corresponding to a layer-3 header (e.g., an IP header), and a segment corresponding to a layer-4 header (e.g., a TCP header).

Pet. 39 (emphases omitted) (quoting Ex. 1007 ¶ 77) (citing Ex. 1003 ¶ 84).

To show how Paatela stores each of multiple headers in “a respective protocol stack layer memory,” Petitioner provides the following highlighted and annotated version of Paatela’s Figure 7 (Pet. 40):

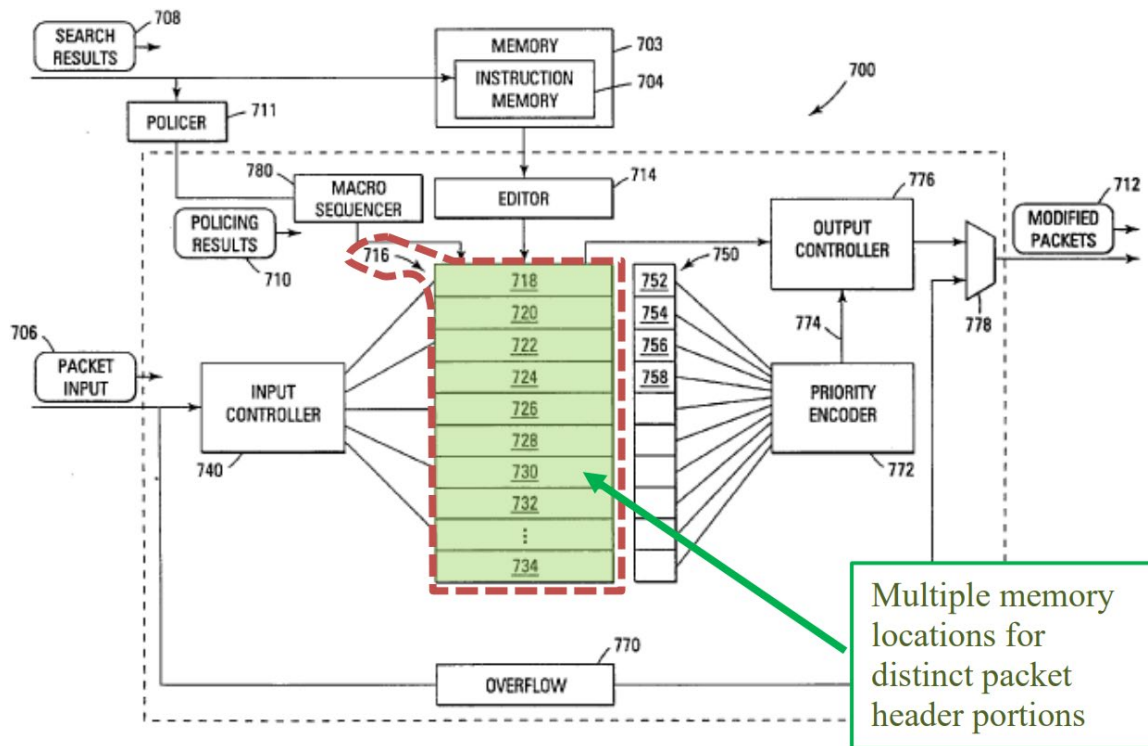


Figure 7 illustrates “an editing system 700” (“also referred to as a packet transformation system”) including (1) editing module 702 (not numbered but identified by a dashed rectangle) with memory 716 comprising a plurality of memory locations 718–734 and (2) memory 703 with instruction memory 704. Ex. 1007 ¶¶ 59–61, 76, Fig. 7; *see id.* ¶ 26. The above highlighted and annotated version of Figure 7 includes:

- (1) a dashed red line around memory 716 comprising a plurality of memory locations 718–734;
- (2) green highlighting over memory locations 718–734; and
- (3) the annotation “Multiple memory locations for distinct header portions” with a green arrow identifying memory locations 718–734.

See Pet. 40; Ex. 1003 ¶ 83.

Petitioner maps the segments of memory locations 718–734 storing “a particular OSI layer header” to the claimed “respective protocol stack layer memory” for header information. Pet. 40 (citing Ex. 1003 ¶ 85). Petitioner asserts that the segments of memory locations 718–734 storing “a particular OSI layer header” store “each individual one of [a] plurality of headers” as claimed. *Id.* (citing Ex. 1003 ¶ 85).

Therefore, according to Petitioner, Paatela teaches “writing . . . each individual one of said plurality of headers to a respective protocol stack layer memory” according to limitation 13.3. Pet. 40 (citing Ex. 1003 ¶ 85).

Patent Owner does not argue that Paatela fails to teach “writing . . . each individual one of said plurality of headers to a respective protocol stack layer memory” according to limitation 13.3. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Paatela teaches “writing . . . each individual one of said plurality of headers to a respective protocol stack layer memory” according to limitation 13.3. *See* Pet. 38–40; Ex. 1003 ¶¶ 83–85.

(iii) Concurrently Writing Each of the Individual Headers

a. Petitioner’s Contentions in the Petition

Next, regarding the requirement for “concurrently writing . . . each individual one of said plurality of headers,” Petitioner contends that Nelson discloses:

- (1) “accelerated packet processing ‘based on the [packet’s] header information without reading the packet out of the buffer’”;

- (2) receive logic 40 receiving packet frames;
- (3) receive logic 40 snooping “on the received data”; and
- (4) receive logic 40 copying the header of each frame to header FIFO 44 “in parallel with the storage of the corresponding frame in frame FIFO 42.”

Pet. 42 (alteration by Petitioner) (emphasis omitted) (quoting Ex. 1006, 1:18–20, 5:7, 7:1–4).

Petitioner asserts that an ordinarily skilled artisan “would have understood Nelson to teach” that (1) “the packet’s header is written” to both “frame FIFO 42 (header and payload)” and “header FIFO 44 (header only)” and (2) the “process is performed in parallel, i.e., concurrently.” Pet. 42 (emphases omitted) (citing Ex. 1003 ¶ 89).

As support, Petitioner provides the following annotated version of Nelson’s Figure 5 (Pet. 43):

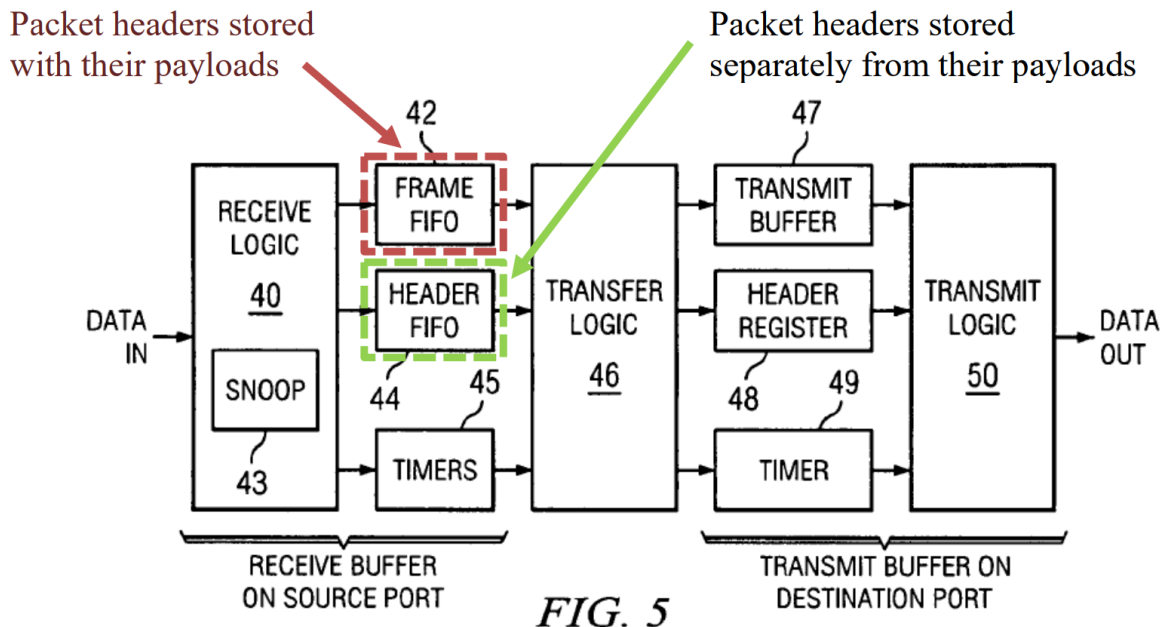


Figure 5 illustrates (1) a receive buffer on a source port including receive logic 40, frame FIFO 42, header FIFO 44, and timers 45, (2) a transmit buffer on a destination port including transmit buffer 47, header register 48,

timer 49, and transmit logic 50, and (3) transfer logic 46 arranged between the receive buffer and the transmit buffer. Ex. 1006, 6:59–7:4, 7:25–39, Fig. 5. The above annotated version of Figure 5 includes:

- (1) a dashed red rectangle around frame FIFO 42;
- (2) the annotation “Packet headers stored with their payloads” with a red arrow identifying frame FIFO 42;
- (3) a dashed green rectangle around header FIFO 44; and
- (4) the annotation “Packet headers stored separately from their payloads” with a green arrow identifying header FIFO 44.

See Pet. 43; Ex. 1003 ¶ 89.

Petitioner also contends that Nelson discloses implementing a buffer with a dual-port random-access memory (RAM) and using a dual-port RAM to store header information instead of a FIFO. Pet. 42 (citing Ex. 1006, 6:8–9); *see id.* at 20 (citing Ex. 1003 ¶¶ 50–51; Ex. 1006, 6:8–9).

Additionally, Petitioner explains that in the Cornett-Paatela-Nelson combination Cornett’s NIC would cause Cornett’s DMA engine to:

- (1) store “the packet (header and payload) in a first buffer” corresponding to the claimed “packet buffer memory”;
- (2) store “each individual packet header in a separate segment of memory” corresponding to the claimed “respective protocol stack layer memory” per Paatela; and
- (3) perform “both storage operations in parallel, i.e., concurrently, per Nelson.”

Pet. 44 (citing Ex. 1003 ¶ 91; Ex. 1005 ¶ 81; Ex. 1006, 6:66–7:4; Ex. 1007 ¶ 77); *see id.* at 32 (citing Ex. 1003 ¶ 67; Ex. 1005 ¶¶ 74, 97; Ex. 1006, 7:1–4; Ex. 1007 ¶ 77).

b. Patent Owner's Contentions in the Response

Patent Owner contends that the Petition's arguments regarding the "concurrently writing" requirement in limitation 13.3 "come up short under the proper construction" because "the Petition only addresses Nelson's purported disclosure of 'concurrently writing' of packet headers to a 'frame FIFO' and to a 'header FIFO.'" Resp. 21 (emphasis omitted). Patent Owner contends that the "Petition does not address how Nelson, or any of the other references for that matter, disclose 'concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.'" *Id.* at 22–23 (emphasis omitted) (citing Ex. 2005 ¶¶ 28–29); *see id.* at 35. Therefore, according to Patent Owner, the "Petition fails to show how Nelson, as part of a combination with Cornett and Paatela, satisfies" the "concurrently writing" requirement in limitation 13.3. *Id.* at 23 (emphases omitted).

According to Patent Owner, the Petition fails to show that claim 13 is unpatentable under § 103(a) because (1) the Petition relies "solely" on Nelson for the "concurrently writing" requirement and (2) Nelson does not teach the "concurrently writing" requirement. Resp. 20, 23. Patent Owner offers two reasons why Nelson does not teach the "concurrently writing" requirement. *See id.* at 23–35.

As the first reason why Nelson does not teach the "concurrently writing" requirement, Patent Owner asserts that:

- (1) "Nelson is generally directed to a system implementation for a Fibre Channel switch";
- (2) "Nelson describes the structure and operation of the Fibre Channel switch with respect to Fig. 5";

- (3) a Fibre Channel frame includes “only a single Fibre Channel header”;
- (4) a Fibre Channel header “is 24-Bytes in length and its fields are defined in the Fibre Channel Specification”;
- (5) the Fibre Channel Specification allows for “optional” Fibre Channel headers;
- (6) any “optional” Fibre Channel headers would reside in the data field of a Fibre Channel frame as “useful data” according to Nelson’s terminology, not in the frame header as “overhead data” according to Nelson’s terminology;
- (7) “Nelson discloses that its Fibre Channel switch does not store ‘useful data’ in the header buffer (header FIFO) because the header buffer (header FIFO) only stores header information, which Nelson distinguishes as ‘overhead data’”; and
- (8) an ordinarily skilled artisan would have recognized that Nelson “only discloses writing a single header to” the header buffer (header FIFO), i.e., writing a single Fibre Channel header, and does not disclose “concurrently writing” the claimed “plurality of headers.”

Resp. 23–31 (emphases omitted) (citing Ex. 1006, 2:13–17, 2:26–39, 3:16–18, 3:26–28, 3:49–57, 4:4–9, 4:34–49, 4:52–60, 5:4–9, 5:60–63, 6:8–12, 6:59–7:4, Figs. 3, 5; Ex. 2005 ¶¶ 31–40; Ex. 2007, 46, 117, 121, 124–46).

As the second reason why Nelson does not teach the “concurrently writing” requirement “even if there was a disclosure in Nelson of multiple headers per frame being written to the header buffer (header FIFO),” Patent Owner asserts that:

- (1) a Fibre Channel “uses a specific type of encoding scheme and requires that all valid transmissions be aligned to a 4-byte word”;

- (2) “consistent with the Fibre Channel Specification’s requirement that valid transmissions be word aligned at 4-Bytes, Nelson’s ‘receive logic’ processes an incoming transmission 4-Bytes (or 4 characters) at a time”;
- (3) a Fibre Channel header “is 24-Bytes in length”;
- (4) for “the 24-Byte Fibre Channel header, 6 sequential cycles would be required to store the entire header in the ‘frame FIFO’ and the ‘header FIFO’”; and
- (5) “Nelson’s ‘receive logic’ does not ‘concurrently write’ the contents of even a single header, let alone the contents of multiple headers.”

Resp. 29, 31–33 (emphases omitted) (quoting Ex. 2005 ¶ 44) (citing Ex. 1006, 6:63–7:1; Ex. 2005 ¶¶ 38, 42–43; Ex. 2007, 124–45).

Additionally, Patent Owner contends that Nelson “confirms the ‘segmented’ storage of header information to the header buffer (header FIFO)” because Nelson “describes the structure of its header buffer (header FIFO) based upon ‘header register elements to store the header information,’ where the ‘header register elements each comprise a chain of cascaded flip-flops . . . which is sufficiently large to hold a frame header.’” Resp. 33 (alteration by Patent Owner) (emphases omitted) (quoting Ex. 1006, 5:60–63) (citing Ex. 2005 ¶ 44).

Patent Owner asserts that an ordinarily skilled artisan would have understood that:

- (1) “a chain of cascaded flip-flops” refers to “a chain of flip-flops where the output of one flip-flop feeds the input of another flip-flop with all the flip-flops being clocked at the same time” (“often referred to as a shift register”);
- (2) Nelson’s disclosure about “a chain of cascaded flip-flops . . . which is sufficiently large to hold a frame header” indicates that (a) “multiple clock cycles are required to

store a single header” and (b) “a single header is stored in the header buffer or header FIFO sequentially”; and

- (3) Nelson “does not disclose ‘concurrently writing’ the single header nor does it disclose ‘concurrently writing’ multiple headers to the header buffer or header FIFO.”

Resp. 33–35 (alteration by Patent Owner) (quoting Ex. 1006, 6:62–63) (citing Ex. 2005 ¶ 44; Ex. 2008, 11, 14).

c. Petitioner’s Contentions in the Reply

In the Reply, Petitioner explains for limitation 13.3 that the Petition relies on:

- (1) Nelson’s disclosure about “simultaneously writing a packet header to a whole-packet memory (i.e., to frame FIFO 42) and to a separate header-only memory (i.e., to header FIFO 44)” to show that the asserted references teach “each of said plurality of headers” being written “‘to a packet buffer memory’ and ‘to a respective protocol stack layer memory’ concurrently”; and
- (2) Cornett’s disclosures and Paatela’s disclosures to show that the asserted references teach other aspects of limitation 13.3.

Reply 8–10 (citing Pet. 36–43; Ex. 1006, 2:3–10, 7:1–4).

Regarding “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other),” Petitioner contends that Nelson discloses using a dual-port memory to store header information instead of a FIFO. Reply 10 (citing Pet. 20, 42; Ex. 1006, 6:8–9); *see* Pet. 20, 42 (quoting Ex. 1006, 6:8–9) (citing Ex. 1003 ¶¶ 50–51). Petitioner contends that Paatela similarly discloses using a dual-port memory “for separately storing individual protocol layer headers.” Reply 10 (citing Ex. 1007 ¶ 78). According to Petitioner, a dual-port memory “can be simultaneously read and/or written by two different data sources, or more

generally, a shared memory accessible by two processes.” *Id.* (emphases omitted) (quoting Ex. 1007 ¶ 78) (citing Ex. 1027, 90:4–10); *see* Pet. 45 (citing Ex. 1006, 6:8–9).

As support, Petitioner quotes Paatela’s disclosure about simultaneously writing multiple “words” to a dual-port memory as follows:

In one embodiment, the data stream is a 64-bit data stream, and the memory 716 is a 32-bit wide, 128-word deep, dual-port memory. In this manner, two 32-bit words may be simultaneously written to the memory 716 to write the 64-bit data. Alternatively, data can be concurrently written to and read from the memory 716.

Reply 10 (emphasis omitted) (quoting Ex. 1007 ¶ 78).

Additionally, Petitioner contends that “Paatela also contemplates using a ‘quad-port memory’ or ‘other multi-port memory.’” Reply 10–11 (citing Ex. 1007 ¶ 78). Petitioner explains that it would have been obvious to an ordinarily skilled artisan that a quad-port memory supports simultaneously writing four 32-bit words since a dual-port memory supports simultaneously writing two 32-bit words. *Id.* at 11 (citing Ex. 1027, 91:16–17). According to Petitioner, it “would have been obvious for Paatela’s ‘multi-port memory’ to have any number of ports, and thus, support simultaneously writing any desired number of 32-bit words.” *Id.* (citing Ex. 1007 ¶ 78).

Further, Petitioner asserts that it “would have been obvious” to an ordinarily skilled artisan for “the simultaneously written 32-bit words to represent multiple different protocol layer headers.” Reply 11. Petitioner explains that “many protocol layer headers are 32 bits long or less.” *Id.* As examples of protocol layer headers that are 32 bits long or less, Petitioner identifies the following headers discussed in Paatela:

- a PPP header, i.e., a layer-2 header, typically comprising 16 to 40 bits; and
- an MPLS label stack, i.e., a layer-2.5 header, typically comprising 32 bits or less.

Id. at 11–12 (citing Ex. 1007 ¶ 77; Ex. 1028, 10:34–37, Fig. 5; Ex. 1029, Fig. 17; Ex. 1030 ¶ 67, Fig. 5A; Ex. 1031 ¶ 58, Fig. 4; Ex. 1032 ¶ 43; Ex. 1033 § 2.1, Fig. 1; Ex. 1034, 6:21–22; Ex. 1035 ¶ 62, Fig. 17A; Ex. 1036 ¶ 21).

Petitioner asserts that “the dual-port memory for headers described by both Nelson and Paatela” supports simultaneously writing two 32-bit protocol headers associated with two different protocols, such as PPP and MPLS. Reply 12. According to Petitioner, simultaneously writing two 32-bit protocol headers associated with two different protocols suffices to satisfy the “construed claim’s recitation” of “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.” *Id.* (emphasis omitted) (citing Ex. 1027, 56:15–57:4).

Petitioner also asserts that Paatela’s disclosure about using a quad-port memory to store headers of the various protocol layers supports simultaneously writing “four 32-bit protocol headers associated with as many as four different protocols.” Reply 12–13 (citing Ex. 1037 ¶ 32). Petitioner further asserts that Paatela’s disclosure about using a multi-port memory with an arbitrary number of ports “would allow an arbitrary number of protocol headers to be simultaneously written.” *Id.* at 13.

Regarding writing multiple headers “each concurrently with each other,” Petitioner contends that:

- (1) “Paatela describes performing simultaneous write operations to the protocol layer-specific header memory 716”;
- (2) “Cornett is ‘dedicated to reducing processing overhead and processing time’”;
- (3) “Paatela is ‘focused on optimizing packet processing in an environment where time is of the essence’”; and
- (4) writing packet data in parallel, as described in Nelson, would facilitate the faster-packet-processing objectives of Cornett and Paatela.

Reply 13 (citing Pet. 39–40; Ex. 1005 ¶ 97; Ex. 1007 ¶¶ 76, 78). Therefore, according to Petitioner, “[w]riting each of the protocol headers to its respective protocol header memory in parallel would have been obvious because doing so would ‘accelerate packet processing.’” *Id.* at 15 (quoting Pet. 32) (citing Ex. 1003 ¶¶ 58, 65–66; Ex. 1006, 7:1–4); *see id.* at 13.

As support, Petitioner quotes Dr. Mir’s testimony as follows:

The combination is premised on the teachings of the references. It permits but does not require physical incorporation of elements from Paatela and Nelson into Cornett. Nevertheless, the physical incorporation of Paatela’s and Nelson’s teachings for how to process individual encapsulated headers of Cornett’s packets, and the timing of writing headers and frames into a buffer, would teach the claim limitations. It would amount to using Paatela’s input controller and packet classification engine to “determine[] where one networking layer header ends and the next networking layer header begins.” Paatela, [0077]. Then, based on this information, using Cornett’s data movement module (e.g., DMA engine) to **move each individual encapsulated header into a respective memory segment**, per Paatela. Cornett, [0074], [0097]. Per Nelson, **the write operations would be performed in parallel to accelerate packet processing**. Nelson, 7:1–4. It would have been obvious to incorporate Paatela’s and Nelson’s teachings into Cornett’s technique.

Reply 14 (alteration in original) (emphases by Petitioner) (quoting Ex. 1003 ¶ 67).

d. Patent Owner's Contentions in the Sur-reply

In the Sur-reply, as in the Response, Patent Owner contends that (1) the Petition relies “solely” on Nelson for the “concurrently writing” requirement and (2) Nelson does not teach the “concurrently writing” requirement. Sur-reply 1–2; *see id.* at 7–9; Resp. 20, 23. In particular, Patent Owner contends that Nelson does not teach “‘concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory,’ as the claims require.” Sur-reply 9; *see id.* at 10.

Patent Owner also contends that “Petitioner’s conclusion that ‘[w]riting each of the protocol headers to its respective protocol header memory in parallel would have been obvious because doing so would “accelerate packet processing”’ is nothing more than unsupported attorney argument and should be disregarded by the Board.” Sur-reply 10 (alteration by Patent Owner) (quoting Reply 15).

As for Petitioner’s reliance on Dr. Mir’s testimony quoted in the Reply, Patent Owner asserts that “these aspects of the declaration were not cited in either the Petition’s, or Dr. Mir’s, analysis of the relevant claim limitation.” Sur-reply 10 (citing Pet. 41–44; Ex. 1003 ¶¶ 88–91). Patent Owner asserts that Petitioner “provides no substantive refutation” to Mr. Brooks’s testimony that Nelson does not teach “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.” *Id.* (emphasis omitted).

Additionally, Patent Owner contends that the Reply includes a new unpatentability theory for the “concurrently writing” requirement based on Paatela’s disclosure about using a dual-port memory, quad-port memory, or multi-port memory to store header information. Sur-reply 2, 11, 14. Patent Owner urges the Board to reject Petitioner’s new unpatentability theory because it is:

- (1) “improper new argument” first raised in the Reply;
- (2) “improper new argument” based on “new prior art regarding the PPP and MPLS headers that should have been cited in the Petition”;
- (3) “attorney argument unsupported by expert testimony”;
and
- (4) deficient on the merits.

Id. at 2, 11–12, 14, 15 n.3.

Regarding an “improper new argument” first raised in the Reply based on Paatela’s disclosure about using a dual-port memory, quad-port memory, or multi-port memory to store header information, Patent Owner asserts that the Petition “never referenced, let alone cited,” that disclosure. Sur-reply 11. According to Patent Owner, “Petitioner’s reply theory crosses the line from the responsive to the new.” *Id.*

Regarding an “improper new argument” based on “new prior art regarding the PPP and MPLS headers that should have been cited in the Petition,” Patent Owner asserts that Petitioner cites nine new references (Exhibits 1028 to 1036) in the Reply for “the proposition that the PPP and MPLS headers are 32-bits long or less.” Sur-reply 15 (citing Reply 10–13). Patent Owner asserts that these nine references “were not cited in the Petition and should be deemed new prior art teaching a claim limitation.”

Id. According to Patent Owner, these nine references “run afoul of the Federal Circuit’s holding ‘that a petitioner may not in reply rely on new prior art to teach a claim limitation.’” *Id.* (quoting *Axonics, Inc. v. Medtronic Inc.*, 75 F.4th 1374, 1383 (Fed. Cir. 2023)).

Regarding “attorney argument unsupported by expert testimony,” Patent Owner asserts that Paatela “makes only a single reference” to a quad-port memory “without further detail as to how that quad-port memory might be used.” Sur-reply 12 (citing Ex. 1007 ¶ 78). According to Patent Owner, Petitioner maintains “without any evidence or explanation” that it would have been obvious to an ordinarily skilled artisan that a quad-port memory supports simultaneously writing four 32-bit words. *Id.* Patent Owner similarly criticizes Petitioner’s contention that a multi-port memory “would allow an arbitrary number of protocol headers to be simultaneously written.” *Id.* Patent Owner asserts that “[t]hese unsupported arguments should be rejected” because “[a]ttorney argument is not evidence.” *Id.* (citing *Elbit Sys. of Am., LLC v. Thales Visionix, Inc.*, 881 F.3d 1354, 1359 (Fed. Cir. 2018)).

Regarding the merits of Petitioner’s argument based on Paatela’s disclosure about using a dual-port memory, quad-port memory, or multi-port memory to store header information, Patent Owner asserts that Paatela does not expressly disclose “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.” Sur-reply 2; *see id.* at 15. Patent Owner asserts that Petitioner’s argument that “the simultaneously written 32-bit words . . . represent multiple different protocol layer headers . . . [if the] headers are 32 bits long or less” is “substantively one of inherency because

there is not an express disclosure of ‘concurrently writing’ a plurality of headers in Paatela.” *Id.* at 16 (alterations by Patent Owner) (emphasis omitted) (quoting Reply 11); *see id.* at 2, 16–17. According to Patent Owner, “inherency requires that a limitation is ‘necessarily present,’ and Paatela does not disclose that ‘concurrently writing’ is anything more than a possibility.” *Id.* at 2 (emphasis omitted); *see id.* at 16.

As another possibility “with respect to Paatela’s disclosure that ‘two 32-bit words may be simultaneously written,’” Patent Owner explains that “the simultaneous writing to both ports of the dual-port memory” could occur “when writing headers that are longer than 32-bits (4-Bytes), such as the 14-Byte Ethernet, 20-Byte IPv4, or 40-Byte IPv6 headers.” Sur-reply 16 n.4 (emphasis omitted) (quoting Ex. 1007 ¶ 78); *see id.* at 17–18.

Additionally, Patent Owner contends that Petitioner takes “inconsistent,” “contradictory,” and “mutually exclusive” positions in the Petition and the Reply. *See* Sur-reply 2, 12–14. Specifically, Patent Owner asserts that the Petition references Nelson’s dual-port RAM “in the context of simultaneous reading from and writing to the dual-port RAM” for the “available for immediate processing within a protocol stack layer” requirement in limitation 13.3. *Id.* at 12–13 (citing Pet. 44–45); *see infra* § III.D.5(d)(iv). As support, Patent Owner quotes the following statement in the Petition: “Nelson’s use of ‘dual port RAM’ to store headers . . . would facilitate immediate processing because data written into dual port RAM can be read out of the RAM at the same time.” Sur-reply 13 (alteration by Patent Owner) (emphasis omitted) (quoting Pet. 45). Patent Owner asserts that the Reply conflicts with the Petition by referencing Nelson’s dual-port RAM for the “concurrently writing” requirement, i.e., to show that “both

ports on a dual-port memory can be written simultaneously to satisfy the claims.” *Id.*

Patent Owner contends that Petitioner “cannot have it both ways because its Reply arguments and Petition arguments are mutually exclusive.” Sur-reply 13. Patent Owner contends that with a dual-port memory there is either:

- (1) “simultaneous writing to both ports on the dual-port memory” to satisfy the “concurrently writing” requirement in limitation 13.3; or
- (2) “simultaneous reading from one port and writing to the other port on the dual-port memory” to satisfy the “available for immediate processing within a protocol stack layer” requirement in limitation 13.3.

Id. at 13–14. In Patent Owner’s view, both requirements “cannot be satisfied.” *Id.* at 14.

e. Analysis

For the reasons explained below, we agree with Petitioner that the combined disclosures in Cornett, Paatela, and Nelson teach “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other)” according to limitation 13.3. *See* Pet. 40–44; Reply 8–15; Ex. 1003 ¶¶ 50–51, 54, 65, 67, 84–85, 88–91; Ex. 3002.

Initially, we disagree with Patent Owner that Petitioner makes an “improper new argument” in the Reply. *See* Sur-reply 2, 11–12, 14. The circumstances here parallel the circumstances in *Axonics, Inc. v. Medtronic, Inc.*, 75 F.4th 1374 (Fed. Cir. 2023).

There, Axonics (the petitioner) did not propose an express construction for any claim term in the petition. *Axonics*, 75 F.4th at 1378. Medtronic (the patent owner) did not propose an express construction for

any claim term in the preliminary response. *Id.* The Board did not expressly construe any claim term in the institution decision. *Id.* Medtronic proposed an express construction for two “wherein” limitations for the first time in the response. *Id.* Then, Axonics argued in the reply that the asserted references disclosed the two “wherein” limitations under Medtronic’s proposed construction. *Id.* at 1369. As support, “Axonics submitted a supplemental expert declaration.” *Id.* Both “the reply and the supplemental expert declaration cited additional disclosures” in the asserted references “pertaining to the same embodiments relied on in the petition.” *Id.* Next, Medtronic argued in the sur-reply that “it would be prejudicial for the Board to consider Axonics’ new reply arguments without providing Medtronic an opportunity to submit its own supplemental expert declaration.” *Id.*

In the final written decision, the Board adopted Medtronic’s proposed construction for the two “wherein” limitations. *Axonics*, 75 F.4th at 1379. Further, the Board “refused to consider” Axonics’ arguments and evidence that the asserted references disclosed the two “wherein” limitations under Medtronic’s proposed construction. *Id.* In particular, the Board “concluded that a reply ‘may not offer an entirely new rationale based on a new combination of elements in the asserted references to show unpatentability based on what amounts to a new ground not set forth in the Petition.’” *Id.*

On appeal, the Federal Circuit decided that “the Board’s refusal to consider the new arguments and evidence was erroneous.” *Axonics*, 75 F.4th at 1376. The Federal Circuit explained that “where a patent owner in an IPR first proposes a claim construction in a patent owner response, a petitioner must be given the opportunity in its reply to argue and present evidence of anticipation or obviousness under the new construction, at least

where it relies on the same embodiments for each invalidity ground as were relied on in the petition.” *Id.* at 1384. The Federal Circuit also noted that there is no rule requiring “a petition to describe all possible or reasonable claim constructions and to present invalidity theories under those constructions.” *Id.* at 1380–81.

Here, as in *Axonics*, Petitioner did not propose an express construction for any claim term in the Petition. *See* Pet. 13; *Axonics*, 75 F.4th at 1378; *supra* § III.C.2. In the Texas case, more than four months after filing the Petition, Petitioner proposed the following construction for the “concurrently writing” requirement: “Plain meaning, i.e., ‘concurrently writing (1) each of said plurality of headers to a CPU main memory space used for the packet transfer and (2) each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.’” Ex. 2001, 4–5 (emphasis omitted); *see* Sur-reply 3–4 (explaining that Petitioner proposed a construction in the Texas case after the deadline for filing a Preliminary Response in this proceeding). In the Texas case, Patent Owner disagreed with Petitioner’s proposed construction and maintained that no construction was necessary. Ex. 2004, 6.

Here, as in *Axonics*, the Board did not expressly construe any claim term in the Institution Decision. *See* Inst. Dec. 6; *Axonics*, 75 F.4th at 1378. Apparently, the parties in the Texas case later agreed upon a construction for the “concurrently writing” requirement because, about two months after the Institution Decision, the district court in the Texas case adopted the following agreed-upon construction for the “concurrently writing” requirement: “Plain meaning, i.e., ‘concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one

of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory.” Ex. 2002, 8.

Here, as in *Axonics*, Patent Owner proposed an express construction for the “concurrently writing” requirement for the first time in the Response. See Resp. 10–18; *Axonics*, 75 F.4th at 1378; *supra* § III.C.3. Specifically, Patent Owner urged the Board to adopt the same construction for the “concurrently writing” requirement as the district court in the Texas case. Resp. 12; *see id.* at 12–18.

Because Patent Owner proposed an express construction for the “concurrently writing” requirement for the first time in the Response, Petitioner “must be given the opportunity in its reply to argue and present evidence of anticipation or obviousness under the new construction, at least where it relies on the same embodiments for each invalidity ground as were relied on in the petition.” See *Axonics*, 75 F.4th at 1384; 37 C.F.R. § 42.23(b).

Here, as in *Axonics*, Petitioner relies on the same embodiments for limitation 13.3 in the Petition and the Reply, e.g., Cornett’s Figure 10 embodiment, Paatela’s Figure 7 embodiment, and Nelson’s Figure 5 embodiment. See Pet. 35–45; Reply 8–15; Ex. 1003 ¶¶ 78–93. Dr. Mir’s testimony quoted in the Reply and quoted above relates to those embodiments and how they “teach the claim limitations.” See Reply 14; Ex. 1003 ¶ 67; *supra* § III.D.5(d)(iii).c.

Although Nelson explains that in “another embodiment,” a dual-port random-access memory (RAM) “can be used to store the header information” instead of a FIFO, Petitioner cites that dual-port RAM embodiment for limitation 13.3 in the Petition. Pet. 42 (citing Ex. 1006,

6:8–9); *see id.* at 20 (citing Ex. 1003 ¶¶ 50–51; Ex. 1006, 6:8–9); Ex. 1006, 6:8–9, 6:16–17, Fig. 4.

In the Reply, Petitioner cites additional disclosures in Paatela about using a dual-port memory, quad-port memory, or multi-port memory as memory 716 to store headers of the various protocol layers. Reply 10–11, 15 (citing Ex. 1007 ¶ 78; Ex. 1027, 90:4–10, 91:16–17). As in *Axonics*, however, those additional disclosures in Paatela pertain to an embodiment Petitioner relied on in the Petition, specifically, Paatela’s Figure 7 embodiment with memory 716 for storing headers of the various protocol layers. Ex. 1007 ¶¶ 60, 76–78, 85, Fig. 7; *see* Pet. 38–40 (citing Ex. 1003 ¶¶ 84–85; Ex. 1007 ¶¶ 60, 76–77, 85, code (57), Fig. 7); Ex. 1003 ¶¶ 83–85.

Further, even if the Federal Circuit’s rule in *Axonics* did not apply, a petitioner “may introduce new evidence after the petition stage if the evidence is a legitimate reply to evidence introduced by the patent owner.” *See Apple Inc. v. Andrea Elecs. Corp.*, 949 F.3d 697, 706 (Fed. Cir. 2020) (“*Andrea*”) (quoting *Anacor Pharms., Inc. v. Iancu*, 889 F.3d 1372, 1380 (Fed. Cir. 2018)); 37 C.F.R. § 42.23(b).

In the Response, Patent Owner asserted that “Nelson’s ‘receive logic’ does not ‘concurrently write’ the contents of even a single header, let alone the contents of multiple headers.” Resp. 33 (emphasis omitted). As support, Patent Owner cited (1) Mr. Brooks’s declaration testimony submitted with the Response and (2) Nelson’s disclosure about using “a chain of cascaded flip-flops” to store header information. *Id.* (citing Ex. 1006, 5:60–63; Ex. 2005 ¶ 44).

In the Reply, Petitioner explained that the Petition cited Nelson’s disclosure about using a dual-port RAM to store header information instead of a FIFO and that Paatela similarly discloses using a dual-port memory to store header information, i.e., “individual protocol layer headers.” Reply 10. In the Reply, Petitioner also explained that Paatela discloses using a quad-port memory or multi-port memory to store header information. *Id.* at 10–11.

The Paatela disclosures cited in the Reply properly address Response arguments because they supplement and expand on the following disclosures cited in the Petition:

- (1) Nelson’s disclosure about using a dual-port RAM to store header information instead of a FIFO; and
- (2) Paatela’s disclosures about Paatela’s Figure 7 embodiment.

See Pet. 20, 22–23, 38–40, 42 (citing Ex. 1003 ¶¶ 50–51, 54, 84; Ex. 1006, 6:8–9; Ex. 1007 ¶¶ 76–77, 85, code (57), Fig. 7); Resp. 31–35; Reply 10–11; Ex. 1006, 6:8–9, 6:16–17; Ex. 1007 ¶¶ 76–78, 85, Fig. 7; *Andrea*, 949 F.3d at 706; *Ericsson Inc. v. Intell. Ventures I LLC*, 901 F.3d 1374, 1379–81 (Fed. Cir. 2018).

Relating to the “concurrently writing” requirement, Paatela explains that memory 716, e.g., a dual-port memory, quad-port memory, or multi-port memory, may be “configured and partitioned such that all header information and data is stored within the memory 716.” Ex. 1007 ¶¶ 78, 85; *see* Ex. 1003 ¶¶ 54, 84. As an example of how memory 716 may store all header information, Paatela explains that:

a segment of memory 716 is allocated to store a layer-2 header (e.g., a PPP header), a segment corresponding to a layer-2.5

header (e.g., an MPLS label stack), a segment corresponding to a layer-3 header (e.g., an IP header), and a segment corresponding to a layer-4 header (e.g., a TCP header).

Ex. 1007 ¶ 77; *see* Ex. 1003 ¶ 84.

Exhibits 1028 to 1036 show that an ordinarily skilled artisan would have known that (1) a PPP header typically comprises 16 to 40 bits (depending on PPP peer negotiation concerning compressing certain fields) and (2) an MPLS label stack typically comprises 32 bits or less. *See* Ex. 1028, 10:30–37, Fig. 5; Ex. 1029 ¶¶ 85, 102, Figs. 12a, 17; Ex. 1030 ¶ 67, Fig. 5A; Ex. 1031 ¶ 58, Fig. 4; Ex. 1032 ¶ 43; Ex. 1033 § 2.1, Fig. 1; Ex. 1034, 6:21–22; Ex. 1035 ¶ 62, Fig. 17A; Ex. 1036 ¶ 21.

Additionally, Nelson discloses storing header information in parallel with corresponding packets or frames of data. Ex. 1006, 2:3–10, 6:29–30, 6:59–7:4, 7:51–57, code (57), Figs. 5–6; *see* Ex. 1003 ¶¶ 58, 65–67, 89–90. In Nelson’s Figure 5 embodiment, for example, receive logic 40 “includes circuitry to snoop on the received data and to copy the header of each frame to header FIFO 44.” Ex. 1006, 6:66–7:1; *see id.* at 7:52–57, Fig. 6; Ex. 1003 ¶ 89. “The copying of a header to header FIFO 44 occurs in parallel with the storage of the corresponding frame in frame FIFO 42.” Ex. 1006, 7:1–4; *see* Ex. 1003 ¶ 89. Hence, “the header information is stored” at “the same time the frame is stored.” Ex. 1006, 6:29–30; *see id.* at 7:52–57, Fig. 6; Ex. 1003 ¶ 89.

Also, as Petitioner explains, in the Cornett-Paatela-Nelson combination Cornett’s NIC would cause Cornett’s DMA engine to:

- (1) store “the packet (header and payload) in a first buffer” corresponding to the claimed “packet buffer memory”;

- (2) store “each individual packet header in a separate segment of memory” corresponding to the claimed “respective protocol stack layer memory” per Paatela; and
- (3) perform “both storage operations in parallel, i.e., concurrently, per Nelson.”

See Pet. 32, 44; Ex. 1003 ¶¶ 67, 89–91; Ex. 1005 ¶¶ 74, 81, 97; Ex. 1006, 2:3–7, 6:66–7:4, Fig. 5; Ex. 1007 ¶¶ 77, 85, Fig. 7.

Where Paatela’s memory 716 comprises a dual-port memory, quad-port memory, or multi-port memory in the Cornett-Paatela-Nelson combination and Paatela’s memory 716 stores, among other things, a 32-bit (or less) PPP header and a 32-bit (or less) MPLS label stack, Paatela teaches that the two 32-bit (or less) headers “may be simultaneously written to the memory 716.” *See* Ex. 1007 ¶¶ 77–78; *see also* Ex. 1003 ¶¶ 67, 91. Because a 32-bit (or less) PPP header and a 32-bit (or less) MPLS label stack correspond to the claimed “plurality of headers,” the combined disclosures in Cornett, Paatela, and Nelson teach “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other).” *See* Ex. 1003 ¶¶ 67, 89–91; Ex. 1005 ¶¶ 74, 81, 97; Ex. 1006, 2:3–7, 6:66–7:4, Fig. 5; Ex. 1007 ¶¶ 77–78, 85, Fig. 7.

As for Patent Owner’s assertion that an ordinarily skilled artisan would have recognized that Nelson “only discloses writing a single header to” the header buffer (header FIFO), i.e., writing a single Fibre Channel header, and does not disclose “concurrently writing” the claimed “plurality of headers,” Petitioner relies on the combined disclosures in Cornett, Paatela, and Nelson to teach the “concurrently writing” requirement. *See* Resp. 30–31; Reply 8–15. As an example, Petitioner contends that the

“combined prior art renders obvious writing multiple layers’ headers to their respective layer memories ‘currently [sic] with each other.’” Reply 13 (emphasis omitted). As another example, Petitioner contends that “the description in Nelson and Paatela of using a dual-port, quad-port, or multi-port memory to store protocol layer headers renders obvious writing multiple layers’ headers ‘currently [sic] with each other’ as required by the claim construction.” *Id.* at 15. Where obviousness rests on a combination of references, a patentee cannot establish nonobviousness by attacking the references individually. *See In re Merck & Co.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

Petitioner relies on Cornett and Paatela—not Nelson—for teaching the claimed “plurality of headers.” Pet. 34, 39–40 (quoting Ex. 1007 ¶ 77). Further, as “specific exemplary protocol headers,” Petitioner identifies “a PPP header” and “an MPLS label stack” as discussed in Paatela instead of a Fibre Channel header as discussed in Nelson. Reply 11 (citing Ex. 1007 ¶ 77); *see* Pet. 39 (quoting Ex. 1007 ¶ 77). Hence, even if a Fibre Channel frame includes “only a single Fibre Channel header,” as Patent Owner asserts, the combined disclosures in Cornett, Paatela, and Nelson still teach the “concurrently writing” requirement. *See* Ex. 1003 ¶¶ 67, 89–91; Ex. 1005 ¶¶ 74, 81, 97; Ex. 1006, 2:3–7, 6:66–7:4, Fig. 5; Ex. 1007 ¶¶ 77–78, 85, Fig. 7; Resp. 28–29; Reply 8–15.

Regarding Nelson’s disclosure of a Fibre Channel fabric formed from multiple interconnected Fibre Channel links and switches, Nelson explains that (1) in “one embodiment, the present system is implemented in a Fibre Channel switch,” (2) the disclosed embodiments are “illustrative,” and (3) “the scope of the invention is not limited to these embodiments.”

Ex. 1006, 3:49–50, 8:25–28; *see* Resp. 28–33. Also, Nelson states that the invention relates to “a system and method implemented in connection with packet switching protocols wherein packet (frame) header information is stored in a buffer separate from the corresponding packet buffer to allow routing decisions to be made based on the header information without reading the packet out of the buffer.” Ex. 1006, 1:13–20; *see id.* at 2:3–10, code (57). Because Nelson’s invention relates generally to packet switching protocols, Patent Owner’s attempt to limit Nelson’s teachings to a Fibre Channel fabric lacks merit. *See* Resp. 28–33.

Regarding Patent Owner’s reliance on Nelson’s disclosure of “header register elements each compris[ing] a chain of cascaded flip-flops,” Petitioner does not propose using “a chain of cascaded flip-flops” in the Cornett-Paatela-Nelson combination. *See* Pet. 24–32, 35–45; Resp. 33–35; Reply 8–15. Instead, Petitioner proposes using “a dual-port, quad-port, or multi-port memory to store protocol layer headers” as disclosed in Paatela and Nelson. Reply 15; *see id.* at 10–15; Pet. 42 (quoting Ex. 1006, 6:8–9).

As for Patent Owner’s assertion that Exhibits 1028 to 1036 “run afoul of the Federal Circuit’s holding ‘that a petitioner may not in reply rely on new prior art to teach a claim limitation,’” Petitioner does not rely on Exhibits 1028 to 1036 to teach a claim limitation. *See* Reply 11–12; Sur-reply 15. Petitioner relies on Exhibits 1028 to 1036 to show that an ordinarily skilled artisan would have known that (1) a PPP header, i.e., a layer-2 header, typically comprises 16 to 40 bits and (2) an MPLS label stack, i.e., a layer-2.5 header, typically comprises 32 bits or less. Reply 11–12. Claim 13 does not specify a header size. Ex. 1001, 12:33–43. Nor does claim 13 require a PPP header or an MPLS label stack. *Id.*

Petitioner properly relies on Exhibits 1028 to 1036 to show the general knowledge of a skilled artisan. *See, e.g., Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1337–38 (Fed. Cir. 2020) (“*Philips*”); *Ariosa Diagnostics v. Verinata Health, Inc.*, 805 F.3d 1359, 1365 (Fed. Cir. 2015); *Randall Mfg. v. Rea*, 733 F.3d 1355, 1362–63 (Fed. Cir. 2013). For example, the Federal Circuit in *Philips* decided that the Board properly determined that challenged claims would have been obvious over a single reference (SMIL 1.0) “in light of the general knowledge of a skilled artisan” as established by expert evidence and a different reference (Hua). *Philips*, 948 F.3d at 1337–38. Further, the Federal Circuit in *Randall* explained that a reliable “form of evidence” to “support a party’s claim about what one of ordinary skill in the relevant art would have known” is “documentary evidence consisting of prior art in the area.” *Randall*, 733 F.3d at 1362–63.

Patent Owner’s assertion that Paatela “makes only a single reference” to a quad-port memory “without further detail as to how that quad-port memory might be used” mischaracterizes Paatela. *See* Sur-reply 12. Paatela explains that memory 716 includes multiple segments or memory locations 718–734 that may store “headers of the various protocol layers associated with the incoming packet.” Ex. 1007 ¶ 77; *see id.* ¶ 85; Ex. 1003 ¶¶ 54, 61, 84. Paatela also explains that memory 716 may comprise “a dual-port memory” that “can be simultaneously read and/or written by two different data sources” such that “two 32-bit words may be simultaneously written to the memory 716” when implemented as a dual-port memory. Ex. 1007 ¶ 78. Then, Paatela discloses that using “a quad-port memory or other multi-port memory will produce analogous results and provide similar advantages” as a dual-port memory. *Id.* Hence, contrary to Patent Owner’s assertion, Paatela

details that “a quad-port memory or other multi-port memory” may serve as memory 716 for storing headers of the various protocol layers. *Id.*

Patent Owner’s assertion that Petitioner maintains “without any evidence or explanation” that it would have been obvious to an ordinarily skilled artisan that a quad-port memory supports simultaneously writing four 32-bit words lacks merit. *See* Sur-reply 12. Paatela discloses that a dual-port memory supports simultaneously writing two 32-bit words, i.e., that “two 32-bit words may be simultaneously written to the memory 716” when implemented as a dual-port memory. Ex. 1007 ¶ 78. Reasoning by analogy, Petitioner explains that it would have been obvious to an ordinarily skilled artisan that a quad-port memory supports simultaneously writing four 32-bit words since a dual-port memory supports simultaneously writing two 32-bit words. Reply 11 (citing Ex. 1027, 91:16–17).

Consistent with Petitioner’s explanation, a 2007 patent application publication discloses a memory device with storage implemented as “a quad-port RAM” having four ports permitting “independent, simultaneous access for read/write from/to any location” within a storage array. Ex. 1037 ¶ 32, Fig. 1. Additionally, Mr. Brooks agrees that “the word ‘quad’ means that there are four.” Ex. 1027, 91:12–17.

As for Patent Owner’s assertion that “[a]ttorney argument is not evidence,” Petitioner cites as evidence supporting its positions various disclosures in Cornett, Paatela, and Nelson, portions of Dr. Mir’s testimony, portions of Mr. Brooks’s testimony, and Exhibits 1028 to 1037. *See* Reply 10–15 (citing Ex. 1003 ¶¶ 58, 65–67, 135; Ex. 1005 ¶ 97; Ex. 1006, 7:1–4; Ex. 1007 ¶¶ 76–78; Ex. 1027, 56:15–57:4, 90:4–10, 91:16–17; Ex. 1028, 10:34–37, Fig. 5; Ex. 1029, Fig. 17; Ex. 1030 ¶ 67, Fig. 5A; Ex. 1031 ¶ 58,

Fig. 4; Ex. 1032 ¶ 43; Ex. 1033 § 2.1, Fig. 1; Ex. 1034, 6:21–22; Ex. 1035 ¶ 62, Fig. 17A; Ex. 1036 ¶ 21; Ex. 1037 ¶ 32).

Moreover, no rule requires “an expert guiding the Board as to how it should read prior art.” *Belden Inc. v. Berk-Tek LLC*, 805 F.3d 1064, 1079 (Fed. Cir. 2015); see *Fanduel, Inc. v. Interactive Games LLC*, 966 F.3d 1334, 1344 (Fed. Cir. 2020). The Board may “independently assess the strength of a petitioner’s argument and evidence.” *Fanduel*, 966 F.3d at 1344.

Patent Owner’s assertion that Petitioner’s argument that “the simultaneously written 32-bit words . . . represent multiple different protocol layer headers . . . [if the] headers are 32 bits long or less” is “substantively one of inherency” lacks merit. See Sur-reply 16. According to Petitioner, it “would have been obvious” to an ordinarily skilled artisan for “the simultaneously written 32-bit words to represent multiple different protocol layer headers.” Reply 11. Petitioner explains that “many protocol layer headers are 32 bits long or less,” such as a PPP header typically comprising 16 to 40 bits and an MPLS label stack typically comprising 32 bits or less. *Id.* at 11–12. Petitioner does not argue that a dual-port memory necessarily supports simultaneously writing two headers or that headers are necessarily 32 bits long or less, and thus does not rely on inherency. See *id.* at 10–15.

That some headers may exceed 32 bits, such as “the 14-Byte Ethernet, 20-Byte IPv4, or 40-Byte IPv6 headers” identified by Patent Owner, does not matter for at least two reasons. See Sur-reply 16 n.4, 17–18. First, prior art that “sometimes meet[s] the claim elements” suffices “to show obviousness.” *Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016). For example, “a prior art product that sometimes, but not

always, embodies a claimed method nonetheless teaches that aspect of the invention.” *Hewlett-Packard Co. v. Mustek Sys., Inc.*, 340 F.3d 1314, 1326 (Fed. Cir. 2003). Second, Paatela discloses using a multi-port memory to store headers of the various protocol layers. Ex. 1007 ¶¶ 77–78, Fig. 7. A multi-port memory with a sufficient number of ports supports simultaneously writing an IPv6 header (or an IPv4 header) and another header, e.g., an Ethernet header. *See* Ex. 1007 ¶ 78; Ex. 1037 ¶ 32.

To support its argument regarding inherency, Patent Owner cites *LBT IP I LLC v. Apple Inc.*, 2023 WL 3914920 (Fed. Cir. June 9, 2023). *See* Ex. 3001. But the circumstances in *LBT* differ materially from the circumstances here.

In *LBT*, a challenged claim included an activation/reactivation limitation. *LBT*, 2023 WL 3914920 at *2. Apple (the petitioner) conceded that the Sakamoto reference did not expressly disclose the activation/reactivation limitation. *Id.* at *3. Apple contended that an ordinarily skilled artisan would have understood that Sakamoto’s system necessarily performed the activation/reactivation limitation under certain operational conditions. *Id.* But the evidence did not establish that those operational conditions necessarily occurred. *Id.* at *3–4.

Here, in contrast to *LBT*, Paatela expressly discloses storing “headers of the various protocol layers associated with the incoming packet” in memory 716, e.g., storing “all header information” in memory 716. Ex. 1007 ¶¶ 77, 85, Fig. 7; *see id.* ¶ 79; Ex. 1003 ¶¶ 54, 61, 84. Paatela expressly discloses using a dual-port memory, quad-port memory, or multi-port memory as memory 716 to store headers of the various protocol layers. Ex. 1007 ¶¶ 77–78, Fig. 7. Further, Paatela expressly discloses that “two

32-bit words may be simultaneously written to the memory 716” when implemented as a dual-port memory. *Id.* ¶ 78. Based on those express disclosures, Petitioner asserts that it “would have been obvious” to an ordinarily skilled artisan for “the simultaneously written 32-bit words to represent multiple different protocol layer headers.” Reply 11. And Petitioner cites competent evidence supporting that assertion. *Id.* at 11–12 (citing Ex. 1007 ¶ 77; Ex. 1028, 10:34–37, Fig. 5; Ex. 1029, Fig. 17; Ex. 1030 ¶ 67, Fig. 5A; Ex. 1031 ¶ 58, Fig. 4; Ex. 1032 ¶ 43; Ex. 1033 § 2.1, Fig. 1; Ex. 1034, 6:21–22; Ex. 1035 ¶ 62, Fig. 17A; Ex. 1036 ¶ 21).

Also, Petitioner relies on the combined disclosures in Cornett, Paatela, and Nelson to teach limitation 13.3, including the “concurrently writing” requirement. *See* Pet. 35–45; Reply 8–15; Ex. 3002. Patent Owner does not identify a limitation missing from the references like the activation/reactivation limitation in *LBT*. *See* Resp. 18–38; Sur-reply 7–21. For these reasons, the circumstances here do not parallel the circumstances in *LBT*.

Additionally, we disagree with Patent Owner that Petitioner takes “inconsistent,” “contradictory,” and “mutually exclusive” positions in the Petition and the Reply regarding Nelson’s dual-port RAM. *See* Sur-reply 2, 12–14. In the Petition, Petitioner asserts for the “concurrently writing” requirement that Nelson’s dual-port RAM “can be used to store the header information” instead of a FIFO. Pet. 42 (citing Ex. 1006, 6:8–9). In the Reply, Petitioner similarly asserts for the “concurrently writing” requirement that “the dual-port memory for headers described by both Nelson and Paatela” supports simultaneously writing two 32-bit protocol headers associated with two different protocols. Reply 12. In the Petition, Petitioner asserts for the “available for immediate processing within a protocol stack

layer” requirement that using Nelson’s dual-port RAM to store the header information would facilitate “immediate processing” because data written to a dual-port RAM “can be read out of the RAM at the same time.” Pet. 45 (citing Ex. 1006, 6:8–9).

We do not view these assertions about Nelson’s dual-port RAM as “inconsistent,” “contradictory,” or “mutually exclusive.” A dual-port memory may support simultaneously writing to different ports, simultaneously reading from different ports, or simultaneously writing to one port and reading from the other port (“‘overlapping’ the input write stage of the processing with the output read stage of the processing”). Ex. 1007 ¶ 78; *see* Ex. 1037 ¶ 32 (discussing “independent, simultaneous access for read/write from/to any location” for a quad-port RAM). Hence, a dual-port memory, such as Nelson’s dual-port RAM, may support:

- (1) simultaneously writing a first header and a second header at an initial time; and
- (2) simultaneously writing a third header and reading an already stored header, e.g., the first header or the second header, at a later time.

Depending on the timing of these memory operations in a dual-port memory, these memory operations are not “mutually exclusive,” contrary to Patent Owner’s contention. *See* Sur-reply 13.

Moreover, Paatela discloses using a quad-port memory or multi-port memory as an alternative to a dual-port memory to “produce analogous results and provide similar advantages” as a dual-port memory. Ex. 1007 ¶ 78. A quad-port memory or multi-port memory supports simultaneously writing two headers and reading an already stored header. *See* Ex. 1037 ¶ 32

(discussing “independent, simultaneous access for read/write from/to any location” for a quad-port RAM).

For the reasons discussed above, the combined disclosures in Cornett, Paatela, and Nelson teach “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other),” and Patent Owner’s arguments to the contrary lack merit.

(iv) Available for Immediate Processing
Within a Protocol Stack Layer

Next, regarding the requirement that each header be “available for immediate processing within a protocol stack layer,” Petitioner asserts that “Paatela explains that packet 706 ‘is stored in the memory 716 in time for the editing instructions from the instruction memory 704 to be processed by the editing processor 714, and for the packet data in the memory 716 to be operated on by the editing processor 714.’” Pet. 44 (emphases omitted) (quoting Ex. 1007 ¶ 76). Petitioner asserts that an ordinarily skilled artisan “would have understood that Paatela’s focus on completing its storage step ‘in time’ for processing teaches that time is of the essence.” *Id.* (citing Ex. 1003 ¶ 92). Petitioner further asserts that using Nelson’s dual-port RAM to store the header information would facilitate “immediate processing” because data written to a dual-port RAM “can be read out of the RAM at the same time.” *Id.* at 45 (citing Ex. 1006, 6:8–9).

Additionally, Petitioner asserts that “Paatela further explains that its editing processor 714 operates on each stored packet header ‘to perform packet modifications and provide packet steering information’ depending on the particular networking protocol.” Pet. 45 (quoting Ex. 1007 ¶ 63).

Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan that in the Cornett-Paatela-Nelson combination each of a packet's headers would be "available for immediate processing within a protocol stack layer," e.g., "even before the payload of the frame (which is generally received after the headers) continues to be written into the frame buffer." Pet. 45 (emphasis omitted) (citing Ex. 1003 ¶ 92; Ex. 1007 ¶ 63); *see id.* at 44 (citing Ex. 1003 ¶ 92).

As discussed above for the "concurrently writing" requirement, Patent Owner contends that with a dual-port memory there is either:

- (1) "simultaneous writing to both ports on the dual-port memory" to satisfy the "concurrently writing" requirement in limitation 13.3; or
- (2) "simultaneous reading from one port and writing to the other port on the dual-port memory" to satisfy the "available for immediate processing within a protocol stack layer" requirement in limitation 13.3.

Sur-reply 13–14; *see supra* § III.D.5(d)(iii).d.

For the reasons stated by Petitioner and supported by Dr. Mir's testimony, we agree with Petitioner that the combined disclosures in Paatela and Nelson teach that each header be "available for immediate processing within a protocol stack layer" according to limitation 13.3. *See* Pet. 44–45; Ex. 1003 ¶¶ 92–93. For the reasons discussed above for the "concurrently writing" requirement, we consider Patent Owner's arguments unavailing. *See supra* § III.D.5(d)(iii).e.

(e) Conclusion Concerning Obviousness/Nonobviousness for Claim 13

For the reasons discussed above, the combined disclosures in Cornett, Paatela, and Nelson teach claim 13's subject matter. *See supra* §§ III.D.5(a)–(d). Moreover, an ordinarily skilled artisan would have been

motivated to combine the teachings of Cornett, Paatela, and Nelson in the way Petitioner proposes and would have had a reasonable expectation of success. *See supra* §§ III.D.4(a)–(b). Hence, Petitioner has shown by a preponderance of the evidence that claim 13 is unpatentable under § 103(a) because the claim would have been obvious over Cornett, Paatela, and Nelson.

6. INDEPENDENT CLAIM 1

(a) Preamble 1.0

Claim 1’s preamble recites “[a] method for processing a packet at an egress end user node.” Ex. 1001, 11:51–52.

Petitioner contends that Cornett teaches claim 1’s preamble because (1) Cornett discloses a node with an NIC corresponding to the claimed “egress end user node” and (2) the NIC performs a “header splitting” process corresponding to the claimed “method for processing a packet.” Pet. 55 (citing Ex. 1003 ¶¶ 113–114); *see id.* at 32.

Patent Owner makes no arguments specific to claim 1’s preamble. *See, e.g.*, Resp. 18–38; Sur-reply 7–21. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

Generally, a preamble does not limit a claim. *Allen Eng’g*, 299 F.3d at 1346. We need not decide whether claim 1’s preamble limits the claim because, for the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches claim 1’s preamble. *See* Pet. 55; Ex. 1003 ¶¶ 113–114.

(b) Limitation 1.1

Claim 1 recites “decoding a packet having a plurality of headers.”
Ex. 1001, 11:53 (limitation 1.1).

Petitioner contends that limitation 1.1 is “substantively identical” to limitation 13.1 and obvious for the same reasons as limitation 13.1, i.e., because Cornett teaches the limitation. Pet. 56; *see id.* at 33–34.

Patent Owner makes no arguments specific to limitation 1.1. *See, e.g.,* Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches limitation 1.1. *See* Pet. 33–34, 56; Ex. 1003 ¶¶ 73–75, 115.

(c) Limitation 1.2

Claim 1 recites “subsequent to said decoding step, concurrently writing (1) each of said plurality of headers to a packet buffer memory and (2) each individual one of said plurality of headers to a respective protocol stack layer memory where it is available for immediate processing within a protocol stack layer.” Ex. 1001, 11:54–59 (limitation 1.2).

Petitioner contends that the combined disclosures in Cornett, Paatela, and Nelson teach limitation 1.2 for the reasons Petitioner contends that the combined disclosures teach limitation 13.3 and because:

- (1) Cornett’s NIC includes “a physical layer device (PHY)” corresponding to the claimed “decoder”;
- (2) Cornett’s NIC “receives signals representing packets from the communications medium”;
- (3) Cornett’s PHY “translates,” i.e., “converts,” the “analog-digital representations of packets into digital bits”; and

- (4) it would have been obvious to an ordinarily skilled artisan that Cornett’s PHY “translates the representations before the digital bits of the packets are transmitted for storage by means of Cornett’s NIC using Cornett’s DMA engine,” e.g., because “it is generally not feasible to directly store an analog signal.”

See Pet. 35–45, 52–53, 56; Reply 8–15.

Patent Owner disputes that the combined disclosures in Cornett, Paatela, and Nelson teach limitation 1.2 for the reasons Patent Owner disputes that the combined disclosures teach limitation 13.3. *See* Resp. 18–35; Sur-reply 7–19.

For the reasons discussed above, we consider Patent Owner’s arguments regarding limitation 13.3 unavailing. *See supra* § III.D.5(d)(iii).e. For (1) the reasons stated by Petitioner and supported by Dr. Mir’s testimony and (2) the reasons discussed above for limitation 13.3, we agree with Petitioner that the combined disclosures in Cornett, Paatela, and Nelson teach limitation 1.2. *See* Pet. 35–45, 52–53, 56; Reply 8–15; Ex. 1003 ¶¶ 78–93, 107–109, 116; *supra* § III.D.5(d)(iii).e.

(d) Conclusion Concerning Obviousness/Nonobviousness for Claim 1

For the reasons discussed above, the combined disclosures in Cornett, Paatela, and Nelson teach claim 1’s subject matter. *See supra* §§ III.D.6(a)–(c). Moreover, an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, and Nelson in the way Petitioner proposes and would have had a reasonable expectation of success. *See supra* §§ III.D.4(a)–(b). Hence, Petitioner has shown by a preponderance of the evidence that claim 1 is unpatentable under § 103(a) because the claim would have been obvious over Cornett, Paatela, and Nelson.

7. DEPENDENT CLAIMS 2, 3, 5–10, 12, 14, 15, AND 17–21

(a) Claims 2 and 14

Claim 2 depends directly from claim 1 and specifies that “at least one of said plurality of headers is a media access control layer protocol header and said respective protocol stack layer memory is a media access control layer memory.” Ex. 1001, 11:60–63. Claim 14 depends directly from claim 13 and includes the same language as claim 2. *Id.* at 12:44–47.

Regarding the “header” requirement in claims 2 and 14, Petitioner asserts that both Cornett and Paatela “operate on TCP/IP packets.” Pet. 46; *see id.* at 56–57. Petitioner also asserts that both Cornett and Paatela “handle Ethernet packets.” *Id.* at 46 (citing Ex. 1005 ¶ 79; Ex. 1007 ¶¶ 79, 113). Petitioner further asserts that an Ethernet header exemplifies “a media access control layer protocol header” as claimed. *Id.* (citing Ex. 1003 ¶ 94; Ex. 1017, 4:2–4; Ex. 1018 ¶ 43; Ex. 1019 ¶ 8). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan that the packets in Cornett and Paatela include “a media access control layer protocol header” as claimed. *Id.* (emphasis omitted) (citing Ex. 1003 ¶ 94).

Regarding the “memory” requirement in claims 2 and 14, Petitioner asserts that Paatela “teaches that a packet frame’s OSI layer 2 header is stored in ‘a segment of memory 716 [that] is allocated to store a layer-2 header (e.g., a PPP header).’” Pet. 46 (alteration by Petitioner) (quoting Ex. 1007 ¶ 77); *see id.* at 56–57. Petitioner also asserts that the “portion of memory segment 716 that stores this information is therefore the claimed media access control layer memory at least because an OSI layer 2 header

includes the MAC header.” *Id.* at 46–47 (emphasis omitted) (citing Ex. 1003 ¶ 95; Ex. 1010 ¶ 48; Ex. 1023, 3).

Further, Petitioner asserts that Paatela “also teaches storing an ‘Ethernet header’ in a specific ‘memory location 718’ separate from other protocol headers” such that “separate memory location 718 alternatively corresponds to the claimed media access control layer memory.” Pet. 47 (emphasis omitted) (citing Ex. 1003 ¶ 95; Ex. 1007 ¶¶ 79, 113, 120). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan “to store a specific protocol layer header in a corresponding layer of memory.” *Id.* (citing Ex. 1003 ¶ 95).

Patent Owner disputes that the references teach the limitations in claims 2 and 14. *See* Sur-reply 2–3, 17–18. Specifically, Patent Owner asserts that:

- (1) claims 2 and 14 require that “at least one of said plurality of headers is a media access control layer protocol header”;
- (2) Petitioner identifies an Ethernet header as the only example of “a media access control layer protocol header” for claims 2 and 14;
- (3) an Ethernet header (excluding the preamble and start of frame delimiter) is 112 bits long (14 bytes long);
- (4) when a PPP header is used within an Ethernet frame, the PPP header is at least 32 bits long (at least 4 bytes long);
- (5) when an MPLS header is used within an Ethernet frame, the MPLS header is 32 bits long (4 bytes long); and
- (6) the combination of Ethernet and PPP/MPLS headers is at least 144 bits long (at least 18 bytes long) and “cannot be simultaneously written” to a 64-bit dual-port memory or a 128-bit quad-port memory.

Id. at 17–18 (citing Pet. 46–47, 56; Ex. 1001, 11:60–62 (claim 2), 12:44–46 (claim 14); Ex. 1029, Figs. 12A–12B; Ex. 1042, 76–78); *see id.* at 2–3.

Therefore, according to Patent Owner, “Paatela cannot satisfy the requirement” of “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory” where “at least one of said plurality of headers is a media access control layer protocol header” according to claims 2 and 14. Sur-reply 18 (emphasis omitted).

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony and the reasons explained in the following paragraph in this section, we agree with Petitioner that the combined disclosures in Cornett and Paatela teach the limitations in claims 2 and 14. *See* Pet. 46–47, 56–57; Ex. 1003 ¶¶ 94–95, 117.

Regarding Patent Owner’s assertions about header lengths relative to a 64-bit dual-port memory and a 128-bit quad-port memory, Paatela discloses using a multi-port memory to store headers of the various protocol layers. Ex. 1007 ¶¶ 77–78, Fig. 7. A multi-port memory with a sufficient number of ports supports simultaneously writing an Ethernet header and another header. *See* Ex. 1007 ¶ 78; Ex. 1037 ¶ 32. Hence, we consider Patent Owner’s assertions unpersuasive.

(b) Claims 3 and 15

Claim 3 depends directly from claim 1 and specifies that “at least one of said plurality of headers is a logic link control layer protocol header and said respective protocol stack layer memory is a logic link control layer memory.” Ex. 1001, 11:64–67. Claim 15 depends directly from claim 13 and includes the same language as claim 3. *Id.* at 12:48–51.

Regarding the “header” requirement in claims 3 and 15, Petitioner asserts that both Cornett and Paatela “operate on TCP/IP packets.” Pet. 47; *see id.* at 56–57. Petitioner also asserts that both Cornett and Paatela “handle Ethernet packets.” *Id.* at 47 (citing Ex. 1005 ¶ 79; Ex. 1007 ¶¶ 79, 113).

Additionally, Petitioner asserts that an ordinarily skilled artisan would have known that an Ethernet frame header may include “a logic link control layer protocol header” as claimed because:

- (1) “‘Ethernet’ commonly refers to both the ‘Ethernet’ networking standard (also known as DIX Ethernet) and to the IEEE 802.3 standard”; and
- (2) the IEEE 802.3 standard requires using a logic link control layer protocol header.

Pet. 47–48 (alteration by Petitioner) (quoting Ex. 1015, 2:14–18) (citing Ex. 1014, 8:42–51, 8:61–64; Ex. 1015, 2:4–7; Ex. 1016, 4:20–28, Figs. 5A–5B); *see* Reply 19–20 (citing Ex. 1014, 8:61–64; Ex. 1015, 2:14–18; Ex. 1042, iii, 3, 31, 42, 97, 249). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan that the Ethernet frames in Cornett and Paatela include “a logic link control layer protocol header” as claimed. Pet. 48 (emphasis omitted) (citing Ex. 1003 ¶ 96); *see* Reply 20–21 (citing Ex. 1043, 6:64–7:23).

Regarding the “memory” requirement in claims 3 and 15, Petitioner asserts that Paatela “teaches that a packet frame’s OSI layer 2 header is stored in ‘a segment of memory 716 [that] is allocated to store a layer-2 header (e.g., a PPP header).’” Pet. 48 (alteration by Petitioner) (quoting Ex. 1007 ¶ 77); *see id.* at 56–57. Petitioner also asserts that the “portion of memory segment 716 that stores this information is therefore the claimed

logic link access control layer memory at least because, in IEEE 802.3 Ethernet, a layer 2 header includes the LLC header.” *Id.* at 48 (emphasis omitted) (citing Ex. 1003 ¶ 97; Ex. 1011, 3:24–26; Ex. 1023, 3).

Further, Petitioner asserts that Paatela “also teaches storing an ‘Ethernet header’ in a specific ‘memory location 718’ separate from other protocol headers” such that “separate memory location 718 alternatively corresponds to the claimed logic link control layer memory.” Pet. 48–49 (emphasis omitted) (citing Ex. 1003 ¶ 97; Ex. 1007 ¶¶ 79, 113, 120). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan “to store a specific protocol layer header in a corresponding layer of memory.” *Id.* at 49 (citing Ex. 1003 ¶ 97).

Patent Owner disputes that the IEEE 802.3 standard requires using a logic link control layer protocol header. *See* Resp. 37; Sur-reply 20–21. Specifically, Patent Owner asserts that “the IEEE 802.3 standard does not provide for conveyance of SNAP and LLC header information.”⁶ Sur-reply 20 (quoting Ex. 1015, 2:31–33).

As support, Patent Owner provides Figure 3-1 (reproduced below) in the 1998 IEEE Standard 802.3:

⁶ The acronym “SNAP” stands for “System Network Architecture Protocol.” Ex. 1015, 2:17–18.

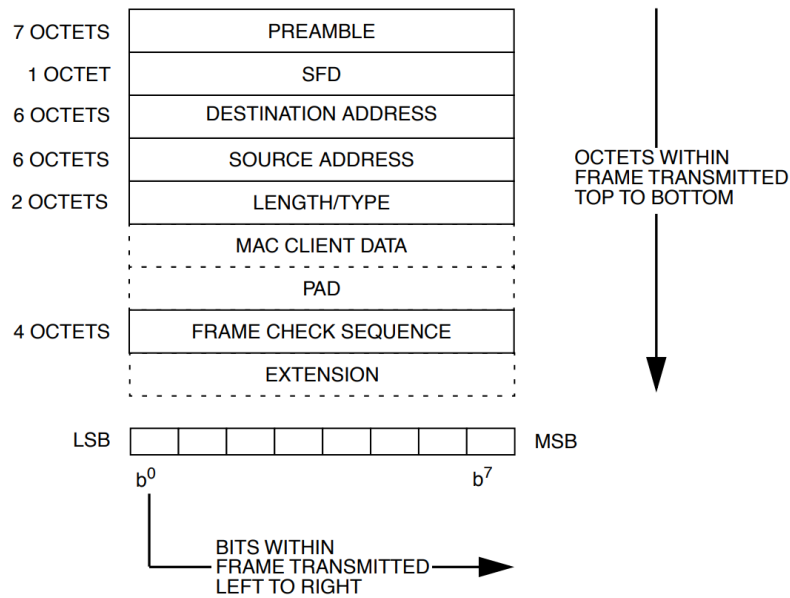


Figure 3-1 — MAC frame format

Figure 3-1 shows the nine fields of a MAC frame, including a 7-octet preamble, a 1-octet start of frame delimiter (SFD), a 6-octet destination address, a 6-octet source address, and a 2-octet length or type field relating to the client data that follows. Ex. 1042, 36.⁷

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony and the reasons explained in the following paragraphs in this section, we agree with Petitioner that the combined disclosures in Cornett and Paatela teach the limitations in claims 3 and 15. *See* Pet. 47–49, 56–57; Reply 19–21; Ex. 1003 ¶¶ 96–97, 118.

As Petitioner asserts, the IEEE 802.3 standard requires using a logic link control layer protocol header. *See* Pet. 48; Reply 19–21. Specifically, the 1998 IEEE Standard 802.3 defines a “data frame” as “the Destination Address, Source Address, Length Field, logical link control (LLC) Data,

⁷ For Exhibit 1042 (1998 IEEE Standard 802.3 Part 3), we follow Petitioner’s practice and cite to the page numbers of the standard instead of the page numbers applied to the exhibit.

PAD, and Frame Check Sequence.” Ex. 1042, 16 (§ 1.4.87). The 1998 IEEE Standard 802.3 does not indicate that the LLC data is optional. *Id.*

Further, Figure 1-1 (reproduced below) in the 1998 IEEE Standard 802.3 depicts the relationship between the seven layers in the OSI reference model and the layers in a local area network (LAN) employing carrier sense multiple access with collision detection (CSMA/CD):

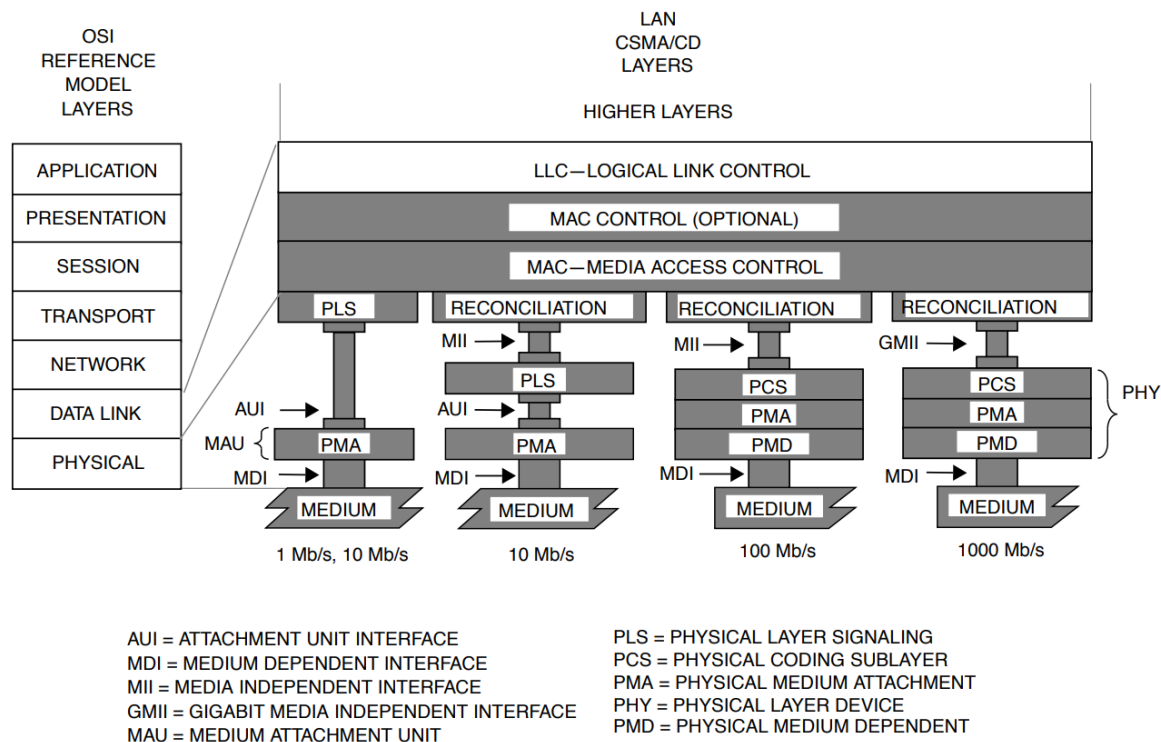


Figure 1-1 — LAN standard relationship to the ISO/IEC Opens Systems Interconnection (OSI) reference model

Figure 1-1 illustrates on the left the OSI reference model layers and on the right the LAN CSMA/CD layers. *See* Ex. 1042, 3. Regarding Figure 1-1, the 1998 IEEE Standard 802.3 explains that the “Logical Link Control (LLC) sublayer and MAC sublayer together encompass the functions intended for the Data Link Layer as defined in the OSI model.” *Id.* at 2. The 1998 IEEE Standard 802.3 does not indicate that the LLC sublayer is

optional. *Id.* at 2–3; *see id.* at 31 (Fig. 2-1a), 42 (Fig. 4-1), 97 (Fig. 6-1), 249 (Fig. 12-1).

Additionally, the patent literature supports Petitioner’s position because:

- a patent issued in 2001 from a 1998 application indicates that an “Ethernet 802.3” frame contains LLC data, including “protocol information” (Ex. 1043, 6:64–7:22);
- a patent issued in 2001 from a 1997 application states that “the 802.3 MAC frame has an IEEE 802.2 logical link control (LLC) header following the MAC header and preceding the OSI network layer protocol header, if one exists” (Ex. 1014, 8:61–64); and
- a patent issued in 2002 from a 2001 application states that some layer-2 protocols, including “802.3 Ethernet,” require “use of a logical link control (LLC) header” (Ex. 1015, 2:14–17).⁸

Hence, we agree with Petitioner (and disagree with Patent Owner) that the IEEE 802.3 standard requires using a logic link control layer protocol header. *See* Pet. 48; Resp. 37; Reply 19–21; Sur-reply 20–21.

As for Patent Owner’s reliance on Figure 3-1 in the 1998 IEEE Standard 802.3, that figure relates to the MAC sublayer and MAC header rather than the LLC sublayer and LLC header. Ex. 1042, 36. Figure 3-1 does not indicate that an LLC sublayer or LLC header is optional. *Id.*

(c) Claims 5 and 17

Claim 5 depends directly from claim 1 and specifies that “at least one of said plurality of headers is a network layer protocol header and said respective protocol stack layer memory is a network layer memory.”

⁸ Petitioner submitted the same patent (US 6,438,128 B1) as Exhibit 1015 and Exhibit 1044.

Ex. 1001, 12:5–8. Claim 17 depends directly from claim 13 and includes the same language as claim 5. *Id.* at 12:56–59.

Regarding the “header” requirement in claims 5 and 17, Petitioner asserts that “the TCP/IP model includes a network layer” that “corresponds to the network layer of the OSI model.” Pet. 49 (quoting Ex. 1010 ¶ 12) (citing Ex. 1010 ¶ 44); *see id.* at 56–57. Petitioner asserts that an ordinarily skilled artisan “would have known that the network layer includes the same information in both the TCP/IP and OSI models.” *Id.* at 49 (citing Ex. 1003 ¶ 98). Petitioner also asserts that Paatela “teaches that an IP header is an example of a header at the network layer (i.e., layer 3) in the OSI model.” *Id.* (citing Ex. 1007 ¶ 77). Therefore, according to Petitioner, an ordinarily skilled artisan “would have known, or it would have been obvious, that the IP header is found in the network layer” and corresponds to “a network layer protocol header” as claimed. *Id.* (emphasis omitted) (citing Ex. 1003 ¶ 99).

Regarding the “memory” requirement in claims 5 and 17, Petitioner asserts that Paatela “teaches that a packet frame’s OSI layer 3 header is stored in ‘a segment of memory 716 [that] is allocated to store a . . . layer-3 header (e.g., an IP header).’” Pet. 49 (alterations by Petitioner) (quoting Ex. 1007 ¶ 77); *see id.* at 56–57. Petitioner also asserts that the “portion of memory segment 716 that stores the IP header (which is an example of a layer 3 header) is the claimed network layer memory.” *Id.* at 50 (emphasis omitted) (citing Ex. 1003 ¶ 100).

Patent Owner disputes that the references teach the limitations in claims 5 and 17. *See* Sur-reply 2–3, 18–19. Specifically, Patent Owner asserts that:

- (1) claims 5 and 17 require that “at least one of said plurality of headers is a network layer protocol header”;
- (2) Petitioner identifies an Internet Protocol header as the only example of “a network layer protocol header” for claims 5 and 17;
- (3) an IPv4 header is 160 bits long (20 bytes long) and exceeds the length of a 64-bit dual-port memory and a 128-bit quad-port memory;
- (4) an IPv6 header is 320 bits long (40 bytes long) and exceeds the length of a 64-bit dual-port memory and a 128-bit quad-port memory; and
- (5) neither an IPv4 header nor an IPv6 header “can be simultaneously written with another header,” such as a PPP/MPLS header, to a 64-bit dual-port memory or a 128-bit quad-port memory.

Id. at 18 (citing Pet. 49–50, 56; Ex. 1001, 12:5–6 (claim 5), 12:56–57 (claim 17); Ex. 1009, 413, 438–39); *see id.* at 2–3.

Therefore, according to Patent Owner, “Paatela cannot satisfy the requirement” of “concurrently writing . . . each individual one of said plurality of headers (each concurrently with each other) to a respective protocol stack layer memory” where “at least one of said plurality of headers is a network layer protocol header” according to claims 5 and 17. Sur-reply 18–19 (emphasis omitted).

Patent Owner also disputes that “the TCP/IP model includes a network layer.” *See* Resp. 36–37; Sur-reply 19–20.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony and the reasons explained in the following paragraphs in this section, we agree with Petitioner that Paatela teaches the limitations in claims 5 and 17. *See* Pet. 49–50, 56–57; Ex. 1003 ¶¶ 98–100, 119.

Regarding Patent Owner’s assertions about header lengths relative to a 64-bit dual-port memory and a 128-bit quad-port memory, Paatela discloses using a multi-port memory to store headers of the various protocol layers. Ex. 1007 ¶¶ 77–78, Fig. 7. A multi-port memory with a sufficient number of ports supports simultaneously writing an IPv6 header (or an IPv4 header) and another header. *See* Ex. 1007 ¶ 78; Ex. 1037 ¶ 32. Hence, we consider Patent Owner’s assertions unpersuasive.

Additionally, Patent Owner’s disagreement that “the TCP/IP model includes a network layer” lacks merit. *See* Resp. 36–37; Sur-reply 19–20. The record establishes that an ordinarily skilled artisan would have known about various models and correlations between layers in those models. *See, e.g.*, Ex. 1009, 35–38;⁹ Ex. 1010 ¶¶ 9–13, 44, Fig. 2; Ex. 1038 ¶ 6, Fig. 2; Ex. 1039, 4:5–23, Fig. 2; Ex. 1040, 3:21–60, Fig. 2; Ex. 1041, 5:49–60, Fig. 3.

For example, a patent issued in 2005 from a 2002 application correlates the seven layers in the OSI reference model to the five layers in the “TCP/IP networking model” as follows:

- (1) the OSI physical layer (layer 1) corresponds to the TCP/IP physical (PHY) layer (layer 1);
- (2) the OSI data link layer (layer 2) corresponds to the TCP/IP media access control (MAC) layer (layer 2);
- (3) the OSI network layer (layer 3) corresponds to the TCP/IP internet (IP) layer (layer 3);

⁹ For Exhibit 1009 (a 1996 textbook titled “Computer Networks”), we follow Petitioner’s practice and cite to the page numbers of the book instead of the page numbers applied to the exhibit.

- (4) the OSI transport layer (layer 4) corresponds to the TCP/IP transport (TCP/UDP) layer (layer 4); and
- (5) the OSI session layer (layer 5), presentation layer (layer 6), and application layer (layer 7) correspond to the TCP/IP application layer (layer 5).

Ex. 1041, 5:49–60, Fig. 3; *see* Ex. 1040, 3:21–60, Fig. 2. Because the OSI network layer (layer 3) corresponds to the TCP/IP internet (IP) layer (layer 3), the TCP/IP model includes a network layer. Ex. 1041, 5:49–60, Fig. 3; *see* Ex. 1040, 3:21–60, Fig. 2.

Even where the TCP/IP model has four layers instead of five layers, the TCP/IP model includes a network layer. *See* Ex. 1009, 35–38; Ex. 1010 ¶¶ 9–13, 44, Fig. 2; Ex. 1038 ¶ 6, Fig. 2. As an example, a 1996 textbook notes that “the TCP/IP internet layer is very similar in functionality to the OSI network layer.” Ex. 1009, 36. As another example, a 2006 patent application publication states that “[t]he TCP/IP network layer corresponds to the network layer of the OSI model.” Ex. 1010 ¶ 12; *see* Ex. 1003 ¶ 98.

Thus, we agree with Petitioner (and disagree with Patent Owner) that “the TCP/IP model includes a network layer.” *See* Pet. 49; Resp. 36–37; Reply 16; Sur-reply 19–20; Ex. 1003 ¶¶ 27, 98.

(d) Claims 6 and 18

Claim 6 depends directly from claim 1 and specifies that “at least one of said plurality of headers is a transport layer protocol header and said respective protocol stack layer memory is a transport layer memory.” Ex. 1001, 12:9–12. Claim 18 depends directly from claim 13 and includes the same language as claim 6. *Id.* at 12:60–63.

Regarding the “header” requirement in claims 6 and 18, Petitioner asserts that “the TCP/IP model includes a transport layer” that “corresponds

to the transport layer of the OSI model.” Pet. 50 (quoting Ex. 1010 ¶ 11) (citing Ex. 1010 ¶ 44); *see id.* at 56–57. Petitioner asserts that an ordinarily skilled artisan “would have known that the transport layer includes the same information in both the TCP/IP and OSI models.” *Id.* at 50 (citing Ex. 1003 ¶ 101). Petitioner also asserts that Paatela “teaches that a TCP header is an example header at the transport layer in the OSI model.” *Id.* (citing Ex. 1007 ¶ 77). Therefore, according to Petitioner, an ordinarily skilled artisan “would have known, or it would have been obvious, that the TCP header is found in the transport layer” and corresponds to “a transport layer protocol header” as claimed. *Id.* (emphasis omitted) (citing Ex. 1001, 7:60–62; Ex. 1003 ¶ 102).

Regarding the “memory” requirement in claims 6 and 18, Petitioner asserts that Paatela “teaches that a packet frame’s OSI layer 4 header is stored in ‘a segment of memory 716 [that] is allocated to store a . . . layer-4 header (e.g., an TCP header).’” Pet. 51 (alterations by Petitioner) (quoting Ex. 1007 ¶ 77); *see id.* at 56–57. Petitioner asserts that (1) the “User Datagram Protocol (UDP) was another well-known transport protocol” and (2) Paatela discloses “storing a UDP header in a separate UDP header memory.” *Id.* at 51 (citing Ex. 1007 ¶¶ 113, 120). Petitioner also asserts that the “portion of memory segment 716 that stores the TCP or UDP header (which are examples of a layer 4 header) is the claimed transport layer memory.” *Id.* (emphasis omitted) (citing Ex. 1003 ¶ 103).

Patent Owner disputes that “the TCP/IP model includes a transport layer.” *See* Resp. 36–37; Sur-reply 19–20.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony and the reasons explained in the following paragraphs in this

section, we agree with Petitioner that Paatela teaches the limitations in claims 6 and 18. *See* Pet. 50–51, 56–57; Ex. 1003 ¶¶ 101–103, 120.

Patent Owner’s disagreement that “the TCP/IP model includes a transport layer” lacks merit. *See* Resp. 36–37; Sur-reply 19–20. As discussed above for claims 5 and 17, the record establishes that an ordinarily skilled artisan would have known about various models and correlations between layers in those models. *See supra* § III.D.7(c). For example, a patent issued in 2005 from a 2002 application correlates the seven layers in the OSI reference model to the five layers in the “TCP/IP networking model” and explains that the OSI transport layer (layer 4) corresponds to the TCP/IP transport (TCP/UDP) layer (layer 4). Ex. 1041, 5:49–60, Fig. 3; *see* Ex. 1040, 3:21–60, Fig. 2; *supra* § III.D.7(c).

Even where the TCP/IP model has four layers instead of five layers, the TCP/IP model includes a transport layer. *See* Ex. 1009, 35–38; Ex. 1010 ¶¶ 9–13, 44, Fig. 2; Ex. 1038 ¶ 6, Fig. 2. For example, a 2006 patent application publication states that “[t]he TCP/IP transport layer corresponds to the transport layer of the OSI model.” Ex. 1010 ¶ 11; *see* Ex. 1003 ¶ 101.

Thus, we agree with Petitioner (and disagree with Patent Owner) that “the TCP/IP model includes a transport layer.” *See* Pet. 50; Resp. 36–37; Reply 16; Sur-reply 19–20; Ex. 1003 ¶¶ 27, 101.

(e) Claims 7 and 19

Claim 7 depends directly from claim 1 and specifies that “at least one of said plurality of headers is an application layer header and said respective protocol stack layer memory is an application layer memory.” Ex. 1001, 12:13–16. Claim 19 depends directly from claim 13 and includes the same language as claim 7. *Id.* at 12:64–67.

Regarding the “header” requirement in claims 7 and 19, Petitioner asserts that “the TCP/IP model includes an application layer” that “corresponds to the combination of the application, presentation and session layers of the OSI model.” Pet. 51 (quoting Ex. 1010 ¶ 10) (citing Ex. 1010 ¶ 11); *see id.* at 56–57. Petitioner asserts that an ordinarily skilled artisan “would have known that the TCP/IP application layer includes the same information as the combination of OSI’s application, presentation, and session layers.” *Id.* at 51 (citing Ex. 1003 ¶ 104). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan “for the application layer to include an application header,” i.e., “an application layer header” as claimed. *Id.* at 51–52 (emphases omitted) (citing Ex. 1003 ¶ 105; Ex. 1013, 2:43–48).

Regarding the “memory” requirement in claims 7 and 19, Petitioner asserts that Paatela “teaches that packets include portions that correspond to ‘various protocol layers’ and packet header information corresponding to OSI networking layers ‘may each correspond to a segment of the memory 716.’” Pet. 52 (quoting Ex. 1007 ¶ 77); *see id.* at 56–57. Petitioner also asserts that Paatela teaches storing headers for multiple layers in memory 716 and “places no limits on which headers it can process.” *Id.* at 52 (citing Ex. 1003 ¶ 106). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan based on Paatela’s teachings “to store and process the application layer header, when present, in a separate portion of memory segment 716” that “corresponds to the claimed application layer memory.” *Id.* (emphasis omitted) (citing Ex. 1003 ¶ 106).

Patent Owner disputes that “the TCP/IP model includes an application layer.” *See* Resp. 36–37; Sur-reply 19–20.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony and the reasons explained in the following paragraphs in this section, we agree with Petitioner that Paatela teaches the limitations in claims 7 and 19. *See* Pet. 51–52, 56–57; Ex. 1003 ¶¶ 104–106, 121.

Patent Owner’s disagreement that “the TCP/IP model includes an application layer” lacks merit. *See* Resp. 36–37; Sur-reply 19–20. As discussed above for claims 5 and 17, the record establishes that an ordinarily skilled artisan would have known about various models and correlations between layers in those models. *See supra* § III.D.7(c). For example, a patent issued in 2005 from a 2002 application correlates the seven layers in the OSI reference model to the five layers in the “TCP/IP networking model” and explains that the OSI session layer (layer 5), presentation layer (layer 6), and application layer (layer 7) correspond to the TCP/IP application layer (layer 5). Ex. 1041, 5:49–60, Fig. 3; *see* Ex. 1003 ¶ 30; Ex. 1040, 3:21–60, Fig. 2; *supra* § III.D.7(c).

Even where the TCP/IP model has four layers instead of five layers, the TCP/IP model includes an application layer. *See* Ex. 1009, 35–38; Ex. 1010 ¶¶ 9–13, 44, Fig. 2; Ex. 1038 ¶ 6, Fig. 2. For example, a 2006 patent application publication states that “[t]he TCP/IP application layer corresponds to a combination of the application, presentation and session layers of the OSI model.” Ex. 1010 ¶ 10; *see* Ex. 1003 ¶ 104.

Thus, we agree with Petitioner (and disagree with Patent Owner) that “the TCP/IP model includes an application layer.” *See* Pet. 51; Resp. 36–37; Reply 16; Sur-reply 19–20; Ex. 1003 ¶¶ 27, 30, 104.

(f) Claim 8

Claim 8 depends directly from claim 1 and further requires “concurrently processing said plurality of headers in each of a plurality of protocol stack layers.” Ex. 1001, 12:17–19.

For claim 8, Petitioner contends that Paatela discloses that “packet 706 ‘is stored in the memory 716 in time for the editing instructions from the instruction memory 704 to be processed by the editing processor 714, and for the packet data in the memory 716 to be operated on by the editing processor 714.’” Pet. 58 (emphases omitted) (quoting Ex. 1007 ¶ 76). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan “for headers to be written to memory in such a way that they become available for immediate processing.” *Id.* (citing Ex. 1003 ¶ 122).

Additionally, Petitioner contends that Paatela discloses that “once the packet’s headers are stored in segments of memory,” the “individual headers are modified as needed,” and then the packet is reassembled with the modified individual headers “in the proper order” such that the modified individual headers effectively replace the original headers. Pet. 58–59 (quoting Ex. 1007 ¶ 79) (citing Ex. 1003 ¶ 123). Petitioner contends that “concurrent processing would have made sense to” an ordinarily skilled artisan to avoid “delayed processing of a header.” *Id.* at 59 (citing Ex. 1003 ¶ 123).

Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan that “Paatela’s processing said plurality of headers would occur concurrently, at least because time is of the essence, and this

processing would occur in each of a plurality of protocol stack layers” as required by claim 8. Pet. 59 (emphases omitted) (citing Ex. 1003 ¶ 124).

Patent Owner makes no arguments specific to claim 8. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Paatela teaches the limitation in claim 8. *See* Pet. 58–59; Ex. 1003 ¶¶ 122–124.

(g) Claims 9 and 20

Claim 9 depends directly from claim 1 and further requires “communicating a portion of said packet to a direct memory access (DMA) device subsequent to said decoding step and prior to said concurrently writing step.” Ex. 1001, 12:20–23. Claim 20 depends directly from claim 13 and includes language similar to claim 9. *Id.* at 13:1–4.

For claims 9 and 20, Petitioner contends that Cornett’s NIC “receives packets and, for those packets selected for header splitting, the NIC ‘cause[s] storage 704 (e.g., via Direct Memory Access (DMA)) of the packet’s header in the page(s) used to store headers and separately store 706 the packet’s payload.” Pet. 52 (alteration by Petitioner) (quoting Ex. 1005 ¶ 74) (citing Ex. 1003 ¶ 107; Ex. 1005 ¶¶ 68, 73); *see id.* at 56, 58. Petitioner contends that Cornett’s NIC includes “a physical layer device (PHY)” corresponding to the claimed “decoder.” *Id.* at 52; *see id.* at 33, 59. Petitioner also contends that Cornett’s PHY is “communicatively coupled to a media access controller (MAC) (e.g., via a FIFO) that performs ‘layer 2’ operations (e.g., Ethernet frame handling).” Pet. 52–53 (quoting Ex. 1005 ¶ 79) (citing Ex. 1003 ¶ 108).

Further, Petitioner asserts that an ordinarily skilled artisan would have understood that (1) Cornett’s NIC “receives signals representing packets from the communications medium” and (2) Cornett’s PHY “translates,” i.e., “converts,” the “analog-digital representations of packets into digital bits.” Pet. 53 (citing Ex. 1003 ¶ 109; Ex. 1005 ¶ 79). Petitioner also asserts that it would have been obvious to an ordinarily skilled artisan that Cornett’s PHY “translates the representations before the digital bits of the packets are transmitted for storage by means of Cornett’s NIC using Cornett’s DMA engine,” e.g., because “it is generally not feasible to directly store an analog signal.” *Id.* (citing Ex. 1003 ¶ 109; Ex. 1005 ¶ 74).

Therefore, according to Petitioner, Cornett renders obvious that its PHY corresponding to the claimed “decoder” is “further configured to communicat[e] a portion of said packet to said DMA device subsequent to decoding said packet” as required by claims 9 and 20. Pet. 53 (alteration by Petitioner) (emphases omitted) (citing Ex. 1003 ¶ 109).

Patent Owner makes no arguments specific to claims 9 and 20. *See, e.g.,* Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches the limitations in claims 9 and 20. *See* Pet. 52–53, 56, 58; Ex. 1003 ¶¶ 107–109, 125.

(h) Claims 10 and 21

Claim 10 depends directly from claim 9 and specifies that “said portion is absent of a preamble and a physical layer header.” Ex. 1001, 12:24–25. Claim 21 depends directly from claim 20 and includes the same language as claim 10. *Id.* at 13:5–6.

For claims 10 and 21, Petitioner asserts that an ordinarily skilled artisan would have understood that:

- (1) the IEEE 802.3 standard defines the structure of an Ethernet packet;
- (2) “a node starts the transmission of [an Ethernet] packet by ‘sending an 8 byte (64 bit) preamble sequence’”;
- (3) the “first 62 bits of this sequence are used to synchronize clocks on the sender and receiver”;
- (4) the “last 2 bits of the sequence are ‘known as the start of frame delimiter’”;
- (5) the “start of frame delimiter is followed by the data communicated at the next-higher layer (e.g., a layer 2 header followed by a layer 2 payload)”;
- (6) “a frame’s physical layer is the lowest layer in both TCP/IP and OSI models.”

Pet. 54 (quoting Ex. 1020, 1:30–31, 1:42–43) (citing Ex. 1003 ¶¶ 110–111; Ex. 1009, 280–81); *see id.* at 56, 58; Reply 17, 22–23.

Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan that “the first 62 bits of the preamble sequence are the claimed preamble and the last 2 bits, known as the start of frame delimiter, are the claimed physical layer header.” Pet. 54 (citing Ex. 1003 ¶ 111).

Further, Petitioner asserts that Cornett discloses that its PHY corresponding to the claimed “decoder” is “communicatively coupled to a media access controller (MAC) (e.g., via a FIFO) that performs ‘layer 2’ operations (e.g., Ethernet frame handling).” Pet. 54 (quoting Ex. 1005 ¶ 79). According to Petitioner, “it was well-known that an Ethernet receive interface ‘starts collecting the bits into bytes for processing by the MAC

layer’ only after the last two bits of the preamble—start of frame delimiter—is received.” *Id.* at 54–55 (quoting Ex. 1020, 1:45–46) (citing Ex. 1003 ¶ 112).

Additionally, Petitioner asserts that an ordinarily skilled artisan would have understood that:

- (1) Cornett’s NIC “receives signals representing packets from the communications medium”;
- (2) Cornett’s PHY “translates,” i.e., “converts,” the “analog-digital representations of packets into digital bits”; and
- (3) Cornett’s PHY “translates the entire packet and then passes the information that follows the preamble and start of frame delimiter for processing.”

Pet. 55 (emphases omitted) (citing Ex. 1003 ¶ 112; Ex. 1005 ¶ 79; Ex. 1020, 1:44–46).

Therefore, according to Petitioner, Cornett renders obvious that its PHY corresponding to the claimed “decoder” transmits “packets (header and payload)—absent of the preamble and physical layer header—using Cornett’s DMA engine for storage” as required by claims 10 and 21. Pet. 55 (emphases omitted) (citing Ex. 1003 ¶ 112; Ex. 1005 ¶ 74); *see* Reply 23.

Patent Owner disputes that the IEEE 802.3 standard defines the structure of an Ethernet packet as including “an 8 byte (64 bit) preamble sequence” as described by Petitioner. Resp. 38; *see* Pet. 54 (referencing the IEEE 802.3 standard). Specifically, Patent Owner asserts that the IEEE 802.3 standard specifies that (1) “a packet’s preamble consist[s] of 7 bytes (56 bits) of a sequence of alternating 1’s and 0’s” and (2) “the preamble was followed by a start of frame delimiter of 1 byte (8 bits) of a 6-bit sequence of

alternating 1's and 0's followed by 2 bits of 1's." Resp. 38 (citing Ex. 2009, 7:11–17).

Patent Owner also disputes that the TCP/IP model includes a physical layer as the lowest layer. *See* Resp. 37; Sur-reply 19–20.

For the reasons stated by Petitioner and supported by Dr. Mir's testimony and the reasons explained in the following paragraphs in this section, we agree with Petitioner that Cornett teaches the limitations in claims 10 and 21. *See* Pet. 54–56, 58; Ex. 1003 ¶¶ 110–112, 126.

Patent Owner's disagreement regarding the structure of an Ethernet packet's preamble has no bearing on the patentability of claims 10 and 21. *See* Resp. 38; Reply 22–23. In particular, the parties agree that:

- (1) an Ethernet packet includes an initial 64-bit sequence comprising a preamble and a start of frame delimiter;
- (2) the first 62 bits of the initial 64-bit sequence consist of alternating 1's and 0's; and
- (3) the last 2 bits of the initial 64-bit sequence are two 1's denoting the sequence's end.

See Pet. 15, 54; Resp. 38; Reply 22.

Consistent with this agreement, the 1998 IEEE Standard 802.3 specifies that the initial 64-bit sequence in an Ethernet packet has the following bit pattern:

10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101011

Ex. 1042, 37, 58. The 1998 IEEE Standard 802.3 explains that the alternating 1's and 0's are “used for physical medium stabilization” and allow the physical signaling (PLS) circuitry to reach steady-state synchronization. *Id.*

The parties disagree whether bit numbers 57 through 62 in the initial 64-bit sequence (the first six positions in the last octet) constitute part of the preamble or part of the start of frame delimiter. *See* Pet. 15, 54–55; Resp. 38; Reply 22–23.

Whether bit numbers 57 through 62 in the initial 64-bit sequence constitute part of the preamble or part of the start of frame delimiter does not matter for claims 10 and 21. Specifically, claims 10 and 21 recite that the portion of a packet communicated to a DMA device “is absent of a preamble and a physical layer header.” Ex. 1001, 12:20–25, 13:1–6. Regarding the claimed “preamble” and the claimed “physical layer header,” Patent Owner does not dispute that:

- (1) an Ethernet packet includes a preamble (whether 56 bits or 62 bits) corresponding to the claimed “preamble”;
- (2) an Ethernet packet includes a start of frame delimiter (whether 8 bits or 2 bits) corresponding to the claimed “physical layer header”; and
- (3) it would have been obvious for Cornett’s PHY to transmit received packet information to Cornett’s DMA engine without including either the preamble (whether 56 bits or 62 bits) or the start of frame delimiter (whether 8 bits or 2 bits).

See Pet. 54–55; Resp. 18–38; Reply 22–23; Sur-reply 7–21; Ex. 1003

¶¶ 110–112. Because an Ethernet packet’s initial 64-bit sequence would be removed entirely, it makes no difference for claims 10 and 21 how the initial 64 bits are divided between a preamble corresponding to the claimed “preamble” and a start of frame delimiter corresponding to the claimed “physical layer header.” *See* Ex. 1003 ¶¶ 110–112; Ex. 1020, 1:30–44; Ex. 2009, 7:4–17.

Additionally, Patent Owner's disagreement that the TCP/IP model includes a physical layer as the lowest layer lacks merit. *See* Resp. 37; Sur-reply 19–20. As discussed above for claims 5 and 17, the record establishes that an ordinarily skilled artisan would have known about various models and correlations between layers in those models. *See supra* § III.D.7(c). For example, a patent issued in 2005 from a 2002 application correlates the seven layers in the OSI reference model to the five layers in the “TCP/IP networking model” and explains that the OSI physical layer (layer 1) corresponds to the TCP/IP physical (PHY) layer (layer 1). Ex. 1041, 5:49–60, Fig. 3; *see* Ex. 1003 ¶ 27; Ex. 1040, 3:21–60, Fig. 2; *supra* § III.D.7(c).

Even where the TCP/IP model has four layers instead of five layers, the TCP/IP model includes a physical layer as the lowest layer. *See* Ex. 1009, 35–38; Ex. 1010 ¶¶ 9–13, 44, Fig. 2; Ex. 1038 ¶ 6, Fig. 2. As an example, a 2005 patent application publication explains that the lowest layer in the TCP/IP model provides “a physical interface with the physical network.” Ex. 1038 ¶ 6. As another example, a 2006 patent application publication explains that the lowest layer in the TCP/IP model includes “network interface card drivers to connect the machine to the physical network, such as an Ethernet network.” Ex. 1010 ¶ 13.

Thus, we agree with Petitioner (and disagree with Patent Owner) that the TCP/IP model includes a physical layer as the lowest layer. *See* Pet. 54; Resp. 37; Reply 16–17; Sur-reply 19–20; Ex. 1003 ¶¶ 27, 111.

(i) Claim 12

Claim 12 depends directly from claim 1 and specifies that “said concurrently writing step is performed within a physical layer utilizing a direct memory access (DMA) device.” Ex. 1001, 12:30–32.

For claim 12, Petitioner contends that Cornett’s NIC includes “a physical layer device (PHY)” corresponding to the claimed “decoder.” Pet. 59 (citing Ex. 1005 ¶ 79); *see id.* at 33, 52. Petitioner also contends that “Cornett’s NIC utiliz[es] the DMA engine, the claimed direct memory access (DMA) device, to write packet headers into memory.” *Id.* at 59 (alteration by Petitioner) (emphases omitted) (citing Ex. 1005 ¶ 74). Therefore, according to Petitioner, it would have been obvious to an ordinarily skilled artisan that “Cornett’s PHY uses Cornett’s DMA engine to perform the concurrently writing step.” *Id.* (emphasis omitted) (citing Ex. 1003 ¶ 127).

Further, Petitioner asserts that it would have been obvious to an ordinarily skilled artisan that “Cornett’s PHY interfaces with the physical layer and higher layers of a protocol stack” because the PHY (1) “receives packets from the communications medium, i.e., the physical layer, translates them, and causes storage of the packet’s headers into memory,” and (2) is “coupled to a media access controller (MAC) . . . that performs ‘layer 2’ operations (e.g., Ethernet frame handling).” Pet. 59–60 (alteration by Petitioner) (quoting Ex. 1005 ¶ 79). Therefore, according to Petitioner, an ordinarily skilled artisan would have understood that “Cornett’s PHY is operating within a physical layer” as required by claim 12. *Id.* at 60 (emphasis omitted) (citing Ex. 1003 ¶ 128).

Patent Owner makes no arguments specific to claim 12. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that Cornett teaches the limitation in claim 12. *See* Pet. 59–60; Ex. 1003 ¶¶ 127–128.

(j) Summary for Dependent Claims 2, 3, 5–10, 12, 14, 15, and 17–21

For the reasons discussed above, the combined disclosures in Cornett, Paatela, and Nelson teach the subject matter in claims 2, 3, 5–10, 12, 14, 15, and 17–21. *See supra* §§ III.D.5(a)–(d), III.D.6(a)–(c), III.D.7(a)–(i). Moreover, an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, and Nelson in the way Petitioner proposes and would have had a reasonable expectation of success. *See supra* §§ III.D.4(a)–(b). Hence, Petitioner has shown by a preponderance of the evidence that claims 2, 3, 5–10, 12, 14, 15, and 17–21 are unpatentable under § 103(a) because the claims would have been obvious over Cornett, Paatela, and Nelson.

*E. Alleged Obviousness over Cornett,
Paatela, Nelson, and Russell: Claim 8*

Petitioner contends that claim 8 is unpatentable under § 103(a) because the claims would have been obvious over Cornett, Paatela, Nelson, and Russell. *See* Pet. 16, 60–69. Patent Owner disputes Petitioner’s contentions. *See* Resp. 1–2, 18–36; Sur-reply 7–19.

Above, we provided overviews of Cornett, Paatela, and Nelson. *See supra* §§ III.D.1–III.D.3. Below, we provide an overview of Russell. Then, we consider the obviousness issues. For the reasons explained below, we

agree with Petitioner that claim 8 is unpatentable under § 103(a) because the claim would have been obvious over Cornett, Paatela, Nelson, and Russell.

1. OVERVIEW OF RUSSELL (EXHIBIT 1025)

Russell is a U.S. patent titled “Processing Network Packets,” filed on August 1, 2000, and issued on January 13, 2004. Ex. 1025, codes (12), (22), (45), (54). Russell states that the invention relates to “systems and methods for processing network packets” that provide “improved packet processing results by logically and physically separating packet header processing functions from packet data processing functions.” *Id.* at 1:6–7, 1:56–58; *see id.* at 4:9–12, code (57). Hence, the invention “enables a network processing system to perform network handling operations and data processing operations substantially in parallel.” *Id.* at 1:59–61; *see id.* at code (57).

Russell’s Figure 3 (reproduced below) depicts a processing system connected to a computer network:

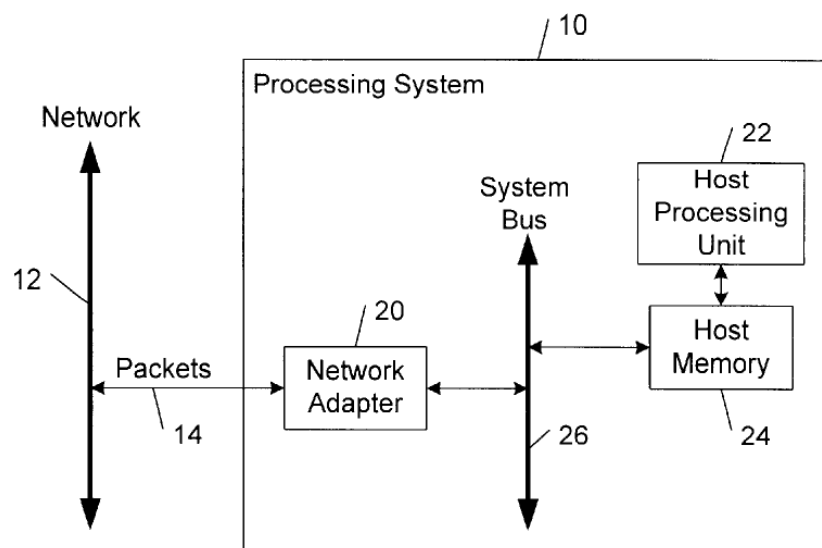


FIG. 3

Figure 3 illustrates processing system 10 connected to computer network 12 where information passes between “processing system 10 and computer network 12 in the form of information packets 14.” Ex. 1025, 2:65–67, 3:13–16, 3:47–50, Fig. 3. Processing system 10 includes network adapter 20, host processing unit 22, host memory 24, and system bus 26. *Id.* at 3:47–51, 3:64–65, Fig. 3.

Network adapter 20 “supports data communication between processing system 10 and network 12.” Ex. 1025, 3:47–50. System bus 26 “couples host processing unit 22 to the various components of processing system 10.” *Id.* at 3:64–65, Fig. 3.

Russell’s Figure 5 (reproduced below) depicts an embodiment of network adapter 20 in more detail:

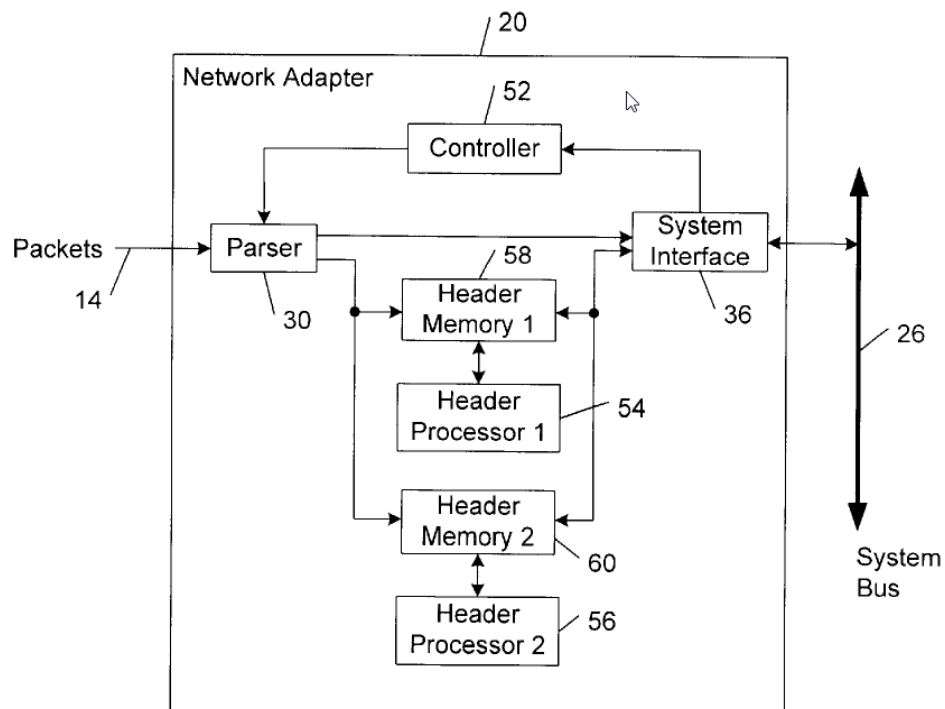


FIG. 5

Figure 5 illustrates network adapter 20 including packet parser 30, system interface 36, controller 52, first header processor 54, second header

processor 56, first header memory 58, and second header memory 60.
Ex. 1025, 3:5, 5:1–7, Fig. 5.

Packet parser 30 “parses a received information packet 14, and directs a portion of the packet header to first header memory 58 and directs the remaining portion of the packet header to second header memory 60.”
Ex. 1025, 5:7–10. Packet parser 30 “also directs the packet data portion of information packet 14 to system interface 36 for transmission to host memory 24 for data processing.” *Id.* at 5:11–13.

First header processor 54 and second header processor 56 “are configured to process packet header information stored in” first header memory 58 and second header memory 60 according to “one or more network protocol handling operations (e.g., firewall filtering, load balancing, and TCP/IP or RTP protocol handling operations).” Ex. 1025, 5:22–27.

Russell explains that the embodiment depicted in Figure 5 permits “packet header processing [to] be performed substantially in parallel by two or more header processors.” Ex. 1025, 5:1–3; *see id.* at 1:56–61, 2:15–21, 8:7–9, code (57).

2. WHETHER AN ORDINARILY SKILLED ARTISAN WOULD
HAVE BEEN MOTIVATED TO COMBINE THE TEACHINGS OF
THE REFERENCES WITH A REASONABLE EXPECTATION OF SUCCESS

(a) Motivation to Combine

Petitioner contends that an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, Nelson, and Russell because “they all aim to accelerate packet processing.” Pet. 62 (citing Ex. 1003 ¶ 133; Ex. 1005 ¶ 97; Ex. 1006, 1:13–20, 2:3–10; Ex. 1007 ¶¶ 76, 79, code (57); Ex. 1025, 1:33–35, 3:13–14, code (57)); *see id.* at 19, 21, 23, 64 (citing Ex. 1001, 1:8–11; Ex. 1003 ¶¶ 48, 52, 55, 136; Ex. 1005

¶ 97; Ex. 1006, 1:13–20, 2:3–10, 3:20–22; Ex. 1007 ¶¶ 41, 76, 79, code (57); Ex. 1025, 1:33–35, 3:13–14, code (57)). Petitioner explains that the combination “would achieve accelerated packet processing that the references seek to achieve.” *Id.* at 65. According to Petitioner, “there is motivation to combine where a known technique ‘has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way.’” *Id.* at 63 (quoting *Intel*, 21 F.4th at 800).

Petitioner contends that Russell reduces latency between “the time a packet is received and the time the packet is processed” by “separating ‘packet header processing functions from packet data processing functions,’” and “then processing header portions ‘substantially in parallel.’” Pet. 64–65 (quoting Ex. 1025, 1:34–35, 8:9, code (57)) (citing Ex. 1003 ¶ 136; Ex. 1025, 3:13–14, 8:8–9). Petitioner also contends that an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, Nelson, and Russell because “Russell’s parallel processing of packet headers would further accelerate packet processing and, as Russell states, reduce latency between the time a packet is received and the time the packet is processed, which would also help attain the goals of Cornett, Paatela, and Nelson.” *Id.* at 68–69 (citing Ex. 1003 ¶ 145; Ex. 1025, 1:33–35).

Further, Petitioner asserts that the Cornett-Paatela-Nelson-Russell combination “does not require physical incorporation of elements from Paatela, Nelson, and Russell into Cornett.” Pet. 66 (citing Ex. 1003 ¶ 139). Although the Cornett-Paatela-Nelson-Russell combination does not require physical incorporation, Petitioner asserts that physical incorporation would have been possible and would have yielded the claimed invention. *Id.*

(citing Ex. 1003 ¶ 139). Petitioner explains that physical incorporation would use:

- (1) “Paatela’s input controller and packet classification engine to ‘determine[] where one networking layer header ends and the next networking layer header begins’”;
- (2) “Cornett’s DMA engine to move each individual packet header into a respective memory segment, per Paatela”;
- (3) a “parallel” move operation for the packet headers per Nelson “to further accelerate packet processing”; and
- (4) “[o]nce the individual packet headers are stored, each would be processed using Russell’s header processors ‘substantially in parallel.’”

Pet. 66–67 (first alteration by Petitioner) (quoting Ex. 1007 ¶ 77; Ex. 1025, 8:9) (citing Ex. 1003 ¶ 139; Ex. 1005 ¶¶ 74, 97; Ex. 1006, 7:1–4).

Additionally, Petitioner contends that the Cornett-Paatela-Nelson-Russell combination “represents the use of known techniques,” specifically:

- (1) “writing packet headers into individual memory segments to accelerate processing, per Paatela”;
- (2) “writing an entire packet frame into one buffer and the packet headers only into another buffer in parallel to accelerate routing decisions, per Nelson”; and
- (3) “processing multiple packet headers in parallel, per Russell.”

Pet. 63 (citing Ex. 1003 ¶ 134). Petitioner contends that using these known techniques would “improve a similar process (accelerated packet processing technique that stores packet headers separately from an entire packet—both header and payload—using a DMA engine, per Cornett) in the same way to yield predictable results.” *Id.* (citing Ex. 1003 ¶ 134).

Patent Owner makes no arguments against combining the teachings of Cornett, Paatela, Nelson, and Russell. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that an ordinarily skilled artisan would have been motivated to combine the teachings of Cornett, Paatela, Nelson, and Russell in the way Petitioner proposes. *See* Pet. 62–69; Ex. 1003 ¶¶ 133–136, 139, 145.

(b) Reasonable Expectation of Success

Petitioner asserts that an ordinarily skilled artisan would have had a reasonable expectation of success in combining Russell’s teachings with the teachings of Cornett, Paatela, and Nelson because the skilled artisan would have appreciated that:

- (1) “Cornett seeks to further accelerate packet processing by using a DMA engine to free up the host processor and move data more efficiently”;
- (2) “Paatela stores individual packet headers in respective, separate memory segments to make each packet header available for processing as quickly as possible”;
- (3) Nelson, like Cornett, “seeks to process packets faster by writing headers and frames into separate buffers—in parallel—so that header information is made available for routing decisions without the need (and associated delay) to read the entire frame from the buffer”;
- (4) “Russell processes headers ‘substantially in parallel’ to further improve packet processing and reduce latency”;
and
- (5) the “combination would achieve accelerated packet processing that the references seek to achieve.”

Pet. 65 (citing Ex. 1003 ¶ 137; Ex. 1005 ¶ 97; Ex. 1006, 2:3–10; Ex. 1007 ¶ 76; Ex. 1025, 1:33–35, 8:8–9, code (57)). Further, Petitioner asserts that an ordinarily skilled artisan “would have been familiar with the concept of processing header portions in parallel.” *Id.* (citing Ex. 1003 ¶ 138); *see id.* at 65–66 (citing Ex. 1024 ¶ 118; Ex. 1026, 3).

Petitioner also asserts that the Cornett-Paatela-Nelson-Russell combination “would have been within the skillset” of an ordinarily skilled artisan because the combination rests on “the teachings of the references themselves.” Pet. 66 (citing Ex. 1003 ¶¶ 137–139); *see id.* at 63 (citing Ex. 1003 ¶ 134).

Patent Owner does not dispute that an ordinarily skilled artisan would have had a reasonable expectation of success in combining the teachings of Cornett, Paatela, Nelson, and Russell. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that an ordinarily skilled artisan would have had a reasonable expectation of success in combining the teachings of Cornett, Paatela, Nelson, and Russell. *See* Pet. 65–67; Ex. 1003 ¶¶ 134, 137–139.

3. DEPENDENT CLAIM 8

Claim 8 depends directly from claim 1 and further requires “concurrently processing said plurality of headers in each of a plurality of protocol stack layers.” Ex. 1001, 12:17–19.

For claim 8, Petitioner contends that Russell discloses “the timing of when each individual packet header stored in a respective, separate segment of memory, per Paatela, is processed” that the Cornett-Paatela-Nelson

combination does not explicitly teach. Pet. 67 (citing Ex. 1003 ¶ 141); *see id.* at 64 (citing Ex. 1003 ¶ 135; Ex. 1025, 1:33–35, 3:13–14, 8:8–9).

Specifically, Petitioner asserts that Russell discloses:

- (1) network adapter 20 including packet parser 30, system interface 36, controller 52, first header processor 54, second header processor 56, first header memory 58, and second header memory 60;
- (2) packet parser 30 parsing “a received information packet 14” and then directing “a portion of the packet header to first header memory 58” and “the remaining portion of the packet header to second header memory 60”;
- (3) first header processor 54 and second header processor 56 processing “packet header information stored in” first header memory 58 and second header memory 60 according to “one or more network protocol handling operations (e.g., firewall filtering, load balancing, and TCP/IP or RTP protocol handling operations)”;
- (4) first header processor 54 and second header processor 56 “each operable to process a respective one or more stored header components substantially in parallel.”

Id. at 67–68 (emphasis omitted) (quoting Ex. 1025, 5:7–10, 5:22–27, 8:7–9) (citing Ex. 1003 ¶¶ 142–144; Ex. 1025, 5:3–7, 8:4–7, Figs. 1, 5).

Further, Petitioner asserts that Cornett, Paatela, and Nelson each seek to “accelerate packet processing.” Pet. 68 (citing Ex. 1005 ¶ 97; Ex. 1006, 2:3–10; Ex. 1007 ¶ 76). Petitioner asserts that “Russell also aims to reduce packet processing latency by ‘logically and physically separat[ing] packet header processing functions from packet data processing functions,’” and “writing packet headers to separate memory locations and processing them ‘substantially in parallel.’” *Id.* (alteration by Petitioner) (quoting Ex. 1025, 5:57–58, 5:61, 8:9, code (57)).

Additionally, Petitioner explains that in the Cornett-Paatela-Nelson-Russell combination:

- (1) “Cornett’s NIC would cause its DMA engine to store the packet (header and payload) in a first buffer, the claimed packet buffer memory, and each individual packet header in a separate segment of memory, the claimed respective protocol stack layer memory, for processing, per Paatela”;
- (2) “Cornett’s NIC would cause the DMA engine to perform both storage operations in parallel, i.e., concurrently, per Nelson”; and
- (3) “[o]nce the individual packet headers are stored, each would be processed ‘substantially in parallel,’ i.e., concurrently, per Russell.”

Pet. 69 (emphases omitted) (citing Ex. 1003 ¶ 146; Ex. 1005 ¶ 81; Ex. 1006, 6:66–7:4; Ex. 1007 ¶ 77; Ex. 1025, 8:8–9).

Therefore, according to Petitioner, an ordinarily skilled artisan would have understood that the Cornett-Paatela-Nelson-Russell combination meets claim 8. Pet. 69 (citing Ex. 1003 ¶¶ 146–147).

Patent Owner makes no arguments specific to claim 8. *See, e.g.*, Resp. 18–38; Sur-reply 7–21.

For the reasons stated by Petitioner and supported by Dr. Mir’s testimony, we agree with Petitioner that the combined disclosures in Cornett, Paatela, Nelson, and Russell teach claim 8’s subject matter. *See* Pet. 67–69; Ex. 1003 ¶¶ 141–147.

4. SUMMARY FOR DEPENDENT CLAIM 8

For the reasons discussed above, the combined disclosures in Cornett, Paatela, Nelson, and Russell teach claim 8’s subject matter. *See supra* §§ III.D.6(a)–(c), III.E.3. Moreover, an ordinarily skilled artisan would have

been motivated to combine the teachings of Cornett, Paatela, Nelson, and Russell in the way Petitioner proposes and would have had a reasonable expectation of success. *See supra* §§ III.E.2(a)–(b). Hence, Petitioner has shown by a preponderance of the evidence that claim 8 is unpatentable under § 103(a) because the claim would have been obvious over Cornett, Paatela, Nelson, and Russell.

IV. CONCLUSION

Based on the evidence presented with the Petition, the evidence introduced during the trial, and the parties’ respective arguments, Petitioner has shown by a preponderance of the evidence that:

- (1) claims 1–3, 5–10, 12–15, and 17–21 are unpatentable under § 103(a) because the claims would have been obvious over Cornett, Paatela, and Nelson; and
- (2) claim 8 is unpatentable under § 103(a) because the claim would have been obvious over Cornett, Paatela, Nelson, and Russell.

In summary:¹⁰

Claim(s)	35 U.S.C. §	Reference(s)/ Basis	Claim(s) Shown Unpatentable	Claim(s) Not Shown Unpatentable
1–3, 5–10, 12–15, 17–21	103(a)	Cornett, Paatela, Nelson	1–3, 5–10, 12–15, 17–21	

¹⁰ Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding after the issuance of this Final Written Decision, we draw Patent Owner’s attention to the April 2019 Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding. *See*

Claim(s)	35 U.S.C. §	Reference(s)/ Basis	Claim(s) Shown Unpatentable	Claim(s) Not Shown Unpatentable
8	103(a)	Cornett, Paatela, Nelson, Russell	8	
Overall Outcome			1–3, 5–10, 12–15, 17–21	

V. ORDER

Accordingly, it is

ORDERED that claims 1–3, 5–10, 12–15, and 17–21 in the
'471 patent are determined to be unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision,
the parties to the proceeding seeking judicial review of the decision must
comply with the notice and service requirements of 37 C.F.R. § 90.2.

84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. *See* 37 C.F.R. § 42.8(a)(3), (b)(2).

IPR2024-01281
Patent 7,738,471 B2

For PETITIONER:

Theodore M. Foster
David L. McCombs
Calmann J. Clements
HAYNES AND BOONE, LLP
ipr.theo.foster@haynesboone.com
david.mccombs.ipr@haynesboone.com
calmann.clements.ipr@haynesboone.com

For PATENT OWNER:

Brett E. Cooper
Seth Lindner
BC LAW GROUP, P.C.
bcooper@bc-lawgroup.com
slindner@bc-lawgroup.com
Lionra_Counsel@b-clg.com

Robert A. Auchter
AUCHTER PLLC
robert@auchterlaw.com