

Cisco Systems, Inc.,
v.
Lionra Technologies Limited

IPR2024-01281, U.S. Patent No. 7,738,471

Patent Owner's Demonstrative Exhibits

December 10, 2025

Ex. 2010

'471 Patent, Claim 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 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.

'471 Patent, Ex. 1001, Claim 13; 12:33-43; *see also* Claim 1, 11:51-59.

Summary of Why The Petition Fails

- The Petition's combination of *Cornett*, *Paatela*, and *Nelson* does not teach independent claims 1 and 13
 - No reference teaches “*concurrently writing ... each individual one of said plurality of headers to a respective protocol stack layer memory*”
- The Reply's new combination of *Cornett*, *Paatela*, and *Nelson* does not teach independent claims 1 and 13
 - Improper new argument based on *Paatela*
 - Contradicts the combination presented in the Petition
 - *Paatela* does not teach “*concurrently writing ... each individual one of said plurality of headers to a respective protocol stack layer memory*”

Paper 12 (POR), 18-36; Paper 18 (Sur-Reply), 7-19.

Ground 1: The Prior Art Does Not Disclose, Teach or Suggest the “Concurrently Writing” Limitation

Construction of “Concurrently Writing ...”

- District Court Agreed Construction: “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.’”

Middle-school grammar rules dictate that a modifier before a numbered list modifies each item in the list. Here, the modifier “concurrently writing” modifies both (1) a 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. And therefore, the claims require that the plurality of headers be written concurrently to each of the respective protocol stack layer memories. This plain-language interpretation is also consistent with each of the embodiments described in the specification, and is required to enable the recited benefits.

Paper 12 (POR), 9-18; Ex. 2004 (Cisco Claim Construction Brief), 1.

Construction of “Concurrently Writing ...” cont’d

The Court need look no further than the plain English language of the claims to resolve this dispute. As is readily apparent, the modifier “concurrently writing” applies to both items in the numbered list. As the Federal Circuit has recognized, it is a “general grammatical principle” that a modifier that comes before a series modifies each item in the series.

* * *

2004)). Thus, a straightforward application of this general grammatical principle dictates that “concurrently writing” also modifies the phrase “each individual one of said plurality of headers to a respective protocol stack layer memory.” In other words, the plain and ordinary meaning requires that the headers must be written each concurrently with each other to their respective protocol stack layer memories.

Ex. 2004 (Cisco Claim Construction Brief), 7 (citations omitted).

Lionra’s recent agreement to a claim construction requiring the concurrent writing of a plurality of headers effectively resolves the matter. Cisco’s products, as evidenced by source code for the UADP, Silicon One, and LSE ASICs, do not write the headers concurrently in the manner described.

Ex. 2003 (April 30, 2025 Letter from Cisco counsel), 1.

Construction of “Concurrently Writing ...” cont’d

In order to solve this problem, a DMA device is used to concurrently write the specific layer header fields directly into: (a) each protocol stack layer’s internal memory spaces **126**, . . . , **134** to initiate immediate processing; and (b) the packet buffer memory **122**.

* * *

The DMA device **120** is also configured to concurrently communicate data 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**. The DMA device **120** allows the transfer of data to these various memory locations in a single DMA transaction.

Ex. 1001 ('471 Patent), 5:64-1, 6:14-21.

Boucher does not teach the step of concurrently writing “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.”⁴

Ex. 1002 ('471 Patent Prosecution History), 30.

Prior Art: Cornett and Paatela

- a. *direct memory access (DMA) device...writing...each of said plurality of headers to a packet buffer memory*

* * *

a POSITA would have understood that Cornett's DMA engine is *configured for [] writing packet headers into a packet buffer memory*. Ex.1003, ¶82.

Paper 2 (Petition), Section XII.E.1.a, 36-38.

- b. *direct memory access (DMA) device...writing... each individual one of said plurality of headers to a respective protocol stack layer memory*

* * *

Cornett teaches storing packet headers in a buffer but does not explicitly disclose how each header is stored or processed. Ex.1003, ¶83. This is disclosed by Paatela.

* * *

Paatela therefore discloses that *each individual one* of a packet's *plurality of headers* is *written* into a separate segment of memory

Paper 2 (Petition), Section XII.E.1.b, 38-41.

Nelson Lacks “Concurrently Writing”

c. *direct memory access (DMA) device...concurrently writing*

The Cornett-Paatela combination *does not explicitly explain the timing* of how Cornett’s packet headers are written to Paatela’s respective, separate segments of memory. This timing is explained in Nelson. Ex.1003, ¶88.

Paper 2 (Petition), Section XII.E.1.c, 41-44.

Nelson teaches 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*. Ex.1006, 7:1-4.

Paper 2 (Petition), 30.

Nelson Lacks “Concurrently Writing”

A POSITA would have

understood Nelson to teach that the packet's header is *written* (1) to a frame FIFO 42 (header and payload) (red outline) and (2) to the header FIFO 44 (header only) (green outline). This process is performed in parallel, i.e., *concurrently*. Ex.1003, ¶89. This is shown in Fig. 5:

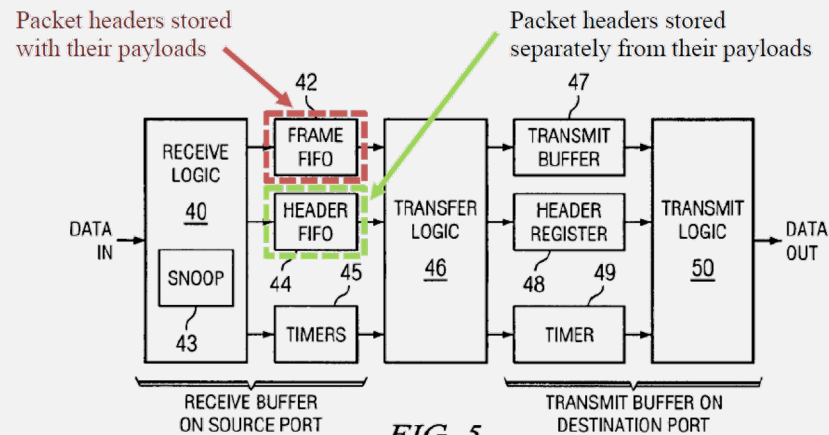


FIG. 5

Paper 2 (Petition), 42-43.

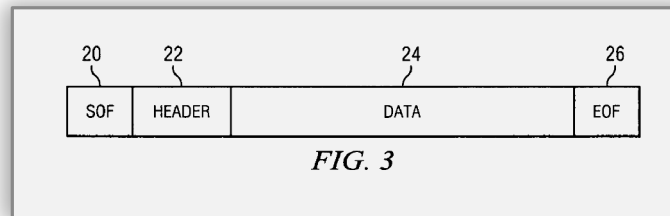
In the combination, Cornett's NIC would cause its DMA engine to store the packet (header and payload) in a first buffer, the claimed *packet buffer memory*, and each individual packet header in a separate segment of memory, the claimed *respective protocol stack layer memory*, for processing, per Paatela. Ex.1005, [0081]; Ex.1007, [0077]; Ex.1003, ¶91. Cornett's NIC would cause the DMA engine to perform both storage operations in parallel, i.e., concurrently, per Nelson.

Paper 2 (Petition), 44.

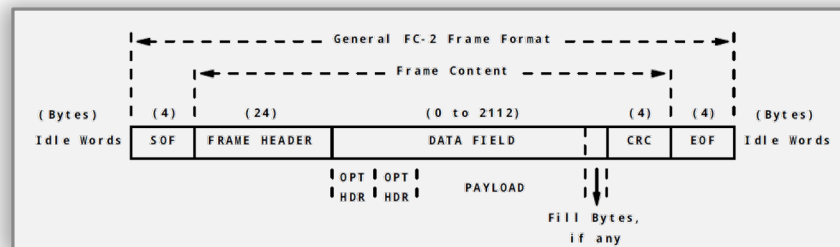
Nelson Uses a Single Header

In one embodiment, the present system is implemented in a Fibre Channel switch. “Fibre Channel” refers to an data

Ex. 1006 (*Nelson*), 3:49-50; see also Ex. 2005 (Brooks Decl.), ¶¶ 31-36.



Ex. 1006 (*Nelson*), Fig. 3.



Ex. 2007 (Fibre Channel Framing and Signaling Specification), Fig. 21.

Nelson Uses Sequential Writing

43. The Nelson patent discloses that the “receive logic is [1] configured to detect the SOF and EOF primitives which define the boundaries of frames ... [2] to store individual frames to locations in frame FIFO 42 ... and [3] to snoop on the received data and to copy the header of each frame to header FIFO 44.” Ex. 1006, 6:63-7:1. A POSITA, understanding this disclosure and the Fibre Channel Specification’s Transmission Word requirement, would recognize that the “receive logic” processes an incoming transmission four characters at a time.

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.” This segmented operation is confirmed by the Nelson patent’s disclosure that “[t]he port snoops on the received data and, **as the header is received**, the port copies the header to a FIFO which serves as header buffer 34.” *Id.*, 5:7-9 (emphasis added).

Ex. 2005 (Brooks Decl.), ¶¶ 43-44; see also Paper 12 (POR), 31-33.

Nelson Uses Sequential Writing

This segmented operation is further confirmed by the Nelson patent's disclosure of "header register elements to store the header information," and where the "header register elements each comprise a *chain of cascaded flip-flops* ... which is sufficiently large to *hold a frame header*." *Id.*, 5:60-63 (emphasis added).

45. The use of "a chain of cascaded flip-flops ... sufficiently large to hold a frame header" indicates to a POSITA that multiple clock cycles are required to store a single header. A POSITA would therefore recognize that a single header is stored in the header buffer or header FIFO sequentially. A POSITA would consequently recognize that the Nelson patent does not disclose "concurrently writing" the single header nor does it disclose "concurrently writing" multiple headers to the header buffer or header FIFO.

Ex. 2005 (Brooks Decl.), ¶¶ 44-45; see also Paper 12 (POR), 33-35.

Petitioner's Reply

- Petitioner does not refute Mr. Brooks' declaration regarding *Nelson*
 - Dr. Mir fails to address *Nelson's* shortcomings as to “concurrently writing”
 - The declaration cites do not address the relevant claim limitation
 - Conclusion of obviousness is unsupported attorney argument

See Paper 18 (Sur-Reply), 10.

Petitioner's Reply cont'd

- Petitioner's new theory is that *Paatela* teaches “concurrently writing”
 - New arguments for the first time in the reply should be rejected
 - No supplemental expert declaration
 - New arguments contradict the Petition
 - *Paatela* does not expressly disclose “concurrently writing”
 - Longer headers of dependent claims 2, 5, 14 and 17 are still not met

See Paper 18 (Sur-Reply), 11-19.

Paatela Reply Theory Unjustified

- Petitioner stated: “Cornett-Paatela combination does not explicitly explain the timing of how Cornett’s packet headers are written to Paatela’s respective, separate segments of memory.” (Paper 2 (Petition), 42)
- In Reply, Petitioner now asserts that *Paatela* does show the timing of how packet headers are written
- Petitioner has no basis for its new theory, based on claim construction or otherwise
 - *Cambridge Mobile Telematics, Inc. v. Sfara, Inc.*, IPR2024-00966, Paper 12 at 8 (PTAB Dec. 13, 2024)

See Paper 18 (Sur-Reply), 3-6, 11-12, 14-15.

Paatela Reply Contradictory

d. *available for immediate processing within a protocol stack layer*

* * *

Nelson's use of "dual port RAM" to store headers, Ex.1006, 6:8-9, would facilitate immediate processing because *data written into dual port RAM can be read out of the RAM at the same time*. It would therefore have been obvious to a POSITA that, in the combination, a packet's headers would be *available for immediate processing* even before the payload of the frame (which is generally received after the headers) continues to be written into the frame buffer. Ex.1003, ¶92.

Paper 2 (Petition), Section XII.E.1.d, 44-45.

Paatela similarly describes using a "dual-port memory" for separately storing individual protocol layer headers. Ex.1007, [0078]. "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." Ex.1007, [0078] (emphasis added); Ex.1027, 90:4-10. Paatela goes on to describe *how multiple "words may be simultaneously written"*:

Paper 15 (Reply), 10-12.

See Paper 18 (Sur-Reply), 12-14.

Paatela Reply Asserts Inherency

- Petitioner’s attorney argument is speculative
 - “Thus, the dual-port memory for headers described by ... *Paatela* **would support** writing two 32-bit protocol headers associated with two different protocols, such as PPP and MPLS.” Paper 15 (Reply), at 12
- Substantively an argument of inherency because *Paatela* undisputedly does not disclose “concurrently writing”
- “Inherency ... may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.”
 - *PAR Pharm., Inc. v. TWI Pharms., Inc.*, 773 F.3d 1186, 1195-96 (Fed. Cir. 2014)

See Paper 18 (Sur-Reply), 15-17.

Paatela Misses Longer Headers in Dependent Claims

- Dependent Claims 2 and 14 requires “at least one of said plurality of headers is a media access control layer protocol header”
 - Petition identified the Ethernet header as the “media access control layer protocol header” (Paper 2 (Petition), at 46-47, 56)
 - Ethernet header length is 112-bits (14-Bytes) (Ex. 1042 (IEEE 802.3), 76-78)
 - When combined with PPP header, *see* Ex. 1029, Figs. 12A, 12B, or MPLS header, *see* Ex. 1033, 3, the combined total of 144-bits (18-Bytes) exceeds the 64-bits (or 128-bits) in Petitioner’s *Paatela* dual-port memory (or quad-port memory) theory
 - Ethernet header cannot be concurrently written to *Paatela*’s protocol layer memory with either PPP or MPLS headers

See Paper 18 (Sur-Reply), 17-18.

Paatela Misses Longer Headers in Dependent Claims

- Dependent Claims 5 and 17 requires “at least one of said plurality of headers is a network layer protocol header”
 - Petition identified the IP header as the “network layer protocol header” (Paper 2 (Petition), at 49-50, 56)
 - IP header length is either 160-bits (20-Bytes) for IPv4, or 320-bits (40-Bytes) for IPv6 (Ex. 1009 (Tanenbaum), 413, 438-439)
 - When combined with PPP or MPLS headers, the combined total of 192-bits (24-Bytes) for IPv4, or 352-bits (44-Bytes) for IPv6, exceeds the 64-bits (or 128-bits) in Petitioner’s *Paatela* dual-port memory (or quad-port memory) theory
 - IP header cannot be concurrently written to *Paatela*’s protocol layer memory with either PPP or MPLS header

See Paper 18 (Sur-Reply), 18-19.

Mr. Brooks – Credible and Unrefuted

- Mr. Brooks applied the agreed construction for “concurrently writing” to the prior art, *i.e.*, *Nelson*
- Petitioner did not attempt to refute Mr. Brooks’ declaration
- Irrelevant cross examination meaningless:
 - History of obsolete standard immaterial
 - Analysis of technologies not at issue in the Petition is pointless
- Nothing inconsistent between Mr. Brooks’ patent and his testimony

See Paper 18 (Sur-Reply), 7-10, 21-24.