



CISCO SYSTEMS, INC.,
Petitioner,
v.
LIONRA TECHNOLOGIES LIMITED,
Patent Owner

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

Theodore M. Foster
Haynes and Boone, LLP

December 10, 2025

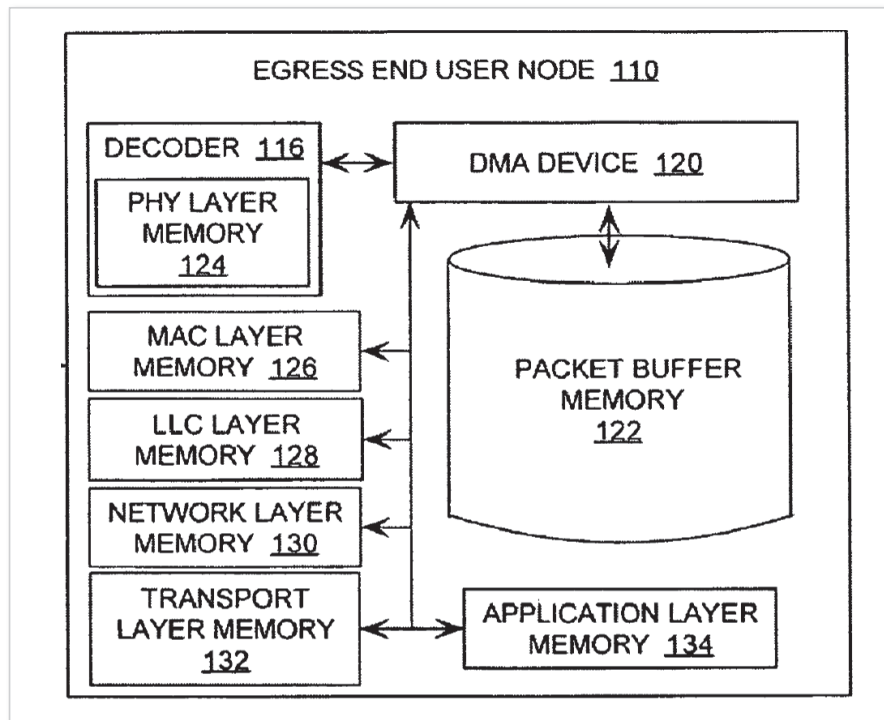


© 2025 Cisco and/or its affiliates. All rights reserved.

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Ex.1050 / IPR2024-01281 / Page 1 of 11
Cisco Systems, Inc. v. Lionra Techs. Ltd.

U.S. Patent No. 7,738,471



Ex.1001, Fig. 1 (partial).

13. An egress end user node (EEUN) of a packet based communications system, comprising:
a decoder configured for decoding a packet having a plurality of headers; and
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.

Ex.1001, Claim 1.

Key Points

1. Claim construction of “*concurrently writing...*”
2. Prior art discloses “*concurrently writing...*” headers to their respective memories

“concurrently writing...”

The Parties agreed to the construction of the following claim term:

Claim Term/Phrase	Agreed Construction
“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” '471 Patent, Claims 1, 13	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 <u>(each concurrently with each other)</u> to a respective protocol stack layer memory.”

(Dkt. No. 45-1 at 2 (P.R. 4-5(d) Joint Claim Construction Chart).) In view of the Parties’ agreement on the proper construction of the identified terms, the Court hereby **ADOPTS** the Parties’ agreed construction.

Ex.2002, 8 (annotated).

Key Points

1. Claim construction of “*concurrently writing...*”
2. Prior art discloses “*concurrently writing...*” headers to their respective memories

“concurrently writing” headers to respective memories

Petition

The Cornett-Paatela-Nelson combination is based on the teachings of the references themselves and does not require physical incorporation of elements from Paatela and Nelson into Cornett. Ex.1003, ¶67. A physical incorporation of the teachings would nevertheless be possible for a POSITA and result in the claimed invention. Ex.1003, ¶67. The physical incorporation would use Paatela’s input controller and packet classification engine to “determine[] where one networking layer header ends and the next networking layer header begins.” Ex.1007, [0077]. It would then use Cornett’s DMA engine to move each individual packet header into a respective memory segment, per Paatela. Ex.1005, [0074], [0097]. The move operation would be performed in parallel, per Nelson, to further accelerate packet processing. Ex.1006, 7:1-4. It would have been obvious to incorporate Paatela’s and Nelson’s teachings into Cornett’s solution. Ex.1003, ¶67.

Paper 2, 31-32.

“concurrently writing” headers to respective memories

Dr. Mir

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.

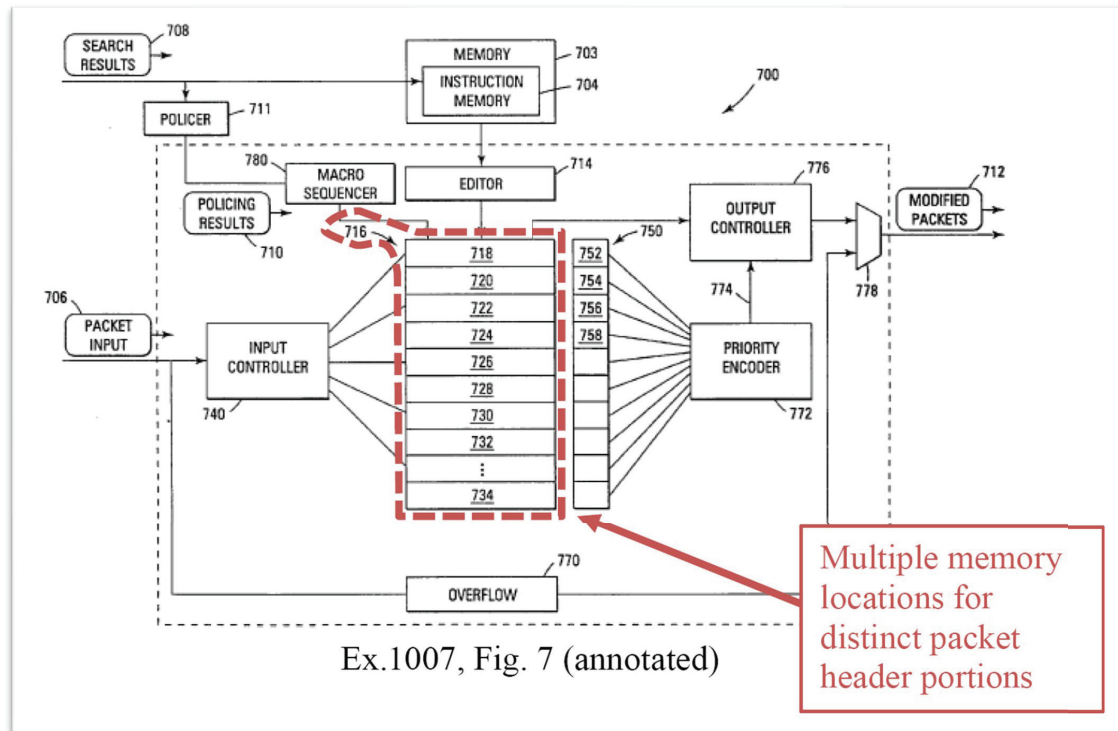
Ex.1003, ¶67 (cited at Paper 2, 31-32; Paper 15, 14).

135. As I explain above in my Section IX.D, the combination of Cornett, Paatela, and Nelson processes packets and uses a DMA engine to save each individual encapsulated header of a packet into a respective memory segment, per Paatela, in parallel, i.e., concurrently, per Nelson. The combination, however, does

Ex.1003, ¶135 (cited at Paper 2, 64; Paper 15, 14-15).

“concurrently writing” headers to respective memories

Paatela



[0078] The memory 716 in one embodiment of the invention is a dual-port memory. 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. 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

Ex.1007, [0078].

Paper 2, 23; see also *id.* 39-40.



© 2025 Cisco and/or its affiliates. All rights reserved.

DEMONSTRATIVE EXHIBIT - NOT EVIDENCE

Ex.1050 / IPR2024-01281 / Page 8 of 11
Cisco Systems, Inc. v. Lionra Techs. Ltd.

“concurrently writing” headers reduces packet latency
(i.e., increases speed)

Nelson

One or more of the problems outlined above may be solved by the various embodiments of the present invention which, broadly speaking, comprises a system and method for storing header information in parallel with corresponding packets or frames of data, wherein the frames of data are stored in a first-in-first-out buffer and wherein the header information is accessed to make routing decisions in order to avoid having to read the frames out of the buffer.

Ex.1006, 2:2-10 (cited at Paper 2, 43).

Combination motivated by desire for speed

Paatela

[0041] Because of the colossal bandwidth demands required of routers, a continual emphasis is placed on alleviating data throughput bottlenecks at routers, gateways, bridges, and other intermediate nodes along the network. 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. The present invention may be used in connection with such routing systems, increasing speed and efficiencies of network data throughput.

Ex.1007, [0041] (cited at Paper 2, 22-23).

Cornett

[0097] Embodiments of the invention may significantly reduce TCP/IP processing overhead that may result from using the core processing module of a host processor. 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. Since the core processing module of a host processor may be bypassed using a DMA engine, slow memory access speeds may be avoided. Furthermore, TCP/IP processing performed on the host processor may scale better than TOE processing because the number of contexts is not limited by TOE memory.

Ex.1005, [0097] (cited at Paper 13, 15).

Dr. Mir

time. Nelson, 6:8-9. A POSITA would thus have been motivated to combine Cornett and Paatela with Nelson because Nelson's parallel writing of headers and frames would further accelerate packet/header processing, which is a stated goal of both Cornett and Paatela.

Ex.1003, 90 (cited at Paper 2, 43-44).



© 2025 Cisco and/or its affiliates. All rights reserved.

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Ex.1050 / IPR2024-01281 / Page 10 of 11
Cisco Systems, Inc. v. Lionra Techs. Ltd.



© 2025 Cisco and/or its affiliates. All rights reserved.

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Ex.1050 / IPR2024-01281 / Page 11 of 11
Cisco Systems, Inc. v. Lionra Techs. Ltd.