NOTE: This disposition is nonprecedential.

United States Court of Appeals for the Federal Circuit

ALACRITECH, INC.,
Appellant

v.

INTEL CORPORATION, CAVIUM, LLC, DELL, INC., WISTRON CORPORATION,
Appellees

UNITED STATES,
Intervenor

2019-1444, 2019-1445, 2019-1466

Appeals from the United States Patent and Trademark Office, Patent Trial and Appeal Board in Nos. IPR2017-01391, IPR2017-01392, IPR2017-01406, IPR2017-01707, IPR2018-00329, IPR2018-00375.

Decided: July 16, 2020

SANFORD IAN WEISBURST, Quinn Emanuel Urquhart & Sullivan, LLP, New York, NY, for appellant. Also represented by JOSEPH M. PAUNOVICH, Los Angeles, CA.

GARLAND STEPHENS, Weil, Gotshal & Manges LLP,

Houston, TX, for appellee Intel Corporation. Also represented by MELISSA LARUE HOTZE; GREGORY SILBERT, New York, NY; AMANDA BRANCH, ANNE MARIE CAPPELLA, Redwood Shores, CA.

KARINEH KHACHATOURIAN, Rimon, P.C., Palo Alto, CA, for appellee Cavium, LLC. Also represented by NIKOLAUS A. WOLOSZCZUK.

KIRK T. BRADLEY, Alston & Bird LLP, Charlotte, NC, for appellee Dell, Inc. Also represented by EMILY CHAMBERS WELCH, Atlanta, GA; BRADY COX, Dallas, TX.

HAROLD H. DAVIS, JR., Greenberg Traurig LLP, San Francisco, CA, for appellee Wistron Corporation. Also represented by YANG LIU, East Palo Alto, CA.

MELISSA N. PATTERSON, Appellate Staff, Civil Division, United States Department of Justice, Washington, DC, for intervenor. Also represented by ETHAN P. DAVIS, COURTNEY DIXON, SCOTT R. MCINTOSH; THOMAS W. KRAUSE, FARHEENA YASMEEN RASHEED, Office of the Solicitor, United States Patent and Trademark Office, Alexandria, VA.

Before MOORE, CHEN, and STOLL, *Circuit Judges.*

CHEN, *Circuit Judge.*

Alacritech appeals from the final written decisions of the Patent Trial and Appeal Board (the Board) in the above-captioned *inter partes* review (IPR) proceedings holding claims 1–7 of U.S. Patent No. 7,237,036, claims 1–

24 of U.S. Patent No. 7,337,241, and claims 1–21 of U.S. Patent No. 7,673,072 invalid as obvious.[1] We *affirm*.

BACKGROUND

The '036, '241, and '072 patents relate to reducing load on a central processing unit (CPU) through offloading certain network processing tasks to a network interface card. Two tasks performed by the network interface card are relevant to this appeal: (1) dividing data into segments for transmission; and (2) updating Transmission Control Protocol (TCP) state information upon receiving a network packet. Claim 1 of the '072 patent and claim 1 of the '036 patent are representative of the data segmentation and TCP state information claims, respectively:

> 1. A method comprising:
>
> establishing, at a host computer, a transport layer connection, including creating a context that includes protocol header information for the connection;
>
> transferring the protocol header information to an interface device;
>
> transferring data from the network host to the interface device, after transferring the protocol header information to the interface device;
>
> *dividing, by the interface device, the data into segments*;

---

[1] Alacritech's appeal briefing included a challenge to the appointment of the administrative patent judges on the Board under the Appointments Clause of the Constitution, but this challenge has since been withdrawn and waived. *See* Dkt. No. 94.

> creating headers for the segments, by the interface device, from a template header containing the protocol header information; and
>
> prepending the headers to the segments to form transmit packets.

'072 patent at claim 1 (emphasis added).

> 1. A device for use with a first apparatus that is connectable to a second apparatus, the first apparatus containing a memory and a first processor operating a stack of protocol processing layers that create a context for communication, the context including a media access control (MAC) layer address, an Internet Protocol (IP) address and Transmission Control Protocol (TCP) state information, the device comprising:
>
> a communication processing mechanism connected to the first processor, said communication processing mechanism containing a second processor running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory and *the TCP state information is updated by said second processor*.

'036 patent at claim 1 (emphasis added).

The Board found all of the challenged claims unpatentable, relying on Erickson[2] and Tanenbaum[3] to disclose the data segmentation and TCP state information limitations. As to the data segmentation claims, the Board explained that Erickson teaches offloading TCP processing from a CPU to a network interface by executing scripts (i.e.,

---

[2] U.S. Patent No. 5,768,618.

[3] ANDREW S. TANENBAUM, COMPUTER NETWORKS (3d ed. 1996).

software instructions) on the network interface to handle the TCP processing. The Board reasoned that, because Tanenbaum discloses that data segmentation is a necessary feature of TCP data transmission, it would have been obvious for Erickson's TCP script to implement data segmentation as a required feature of TCP. As to the claims on updating TCP state information, the Board agreed with the petitioner that it would have been obvious for Erickson's script, executed by the network interface, to update connection records when receiving packets as taught by Tanenbaum. The Board found that updating the connection records would involve updating TCP state information such as TCP sequence numbers.

Alacritech appeals, and we have jurisdiction under 28 U.S.C. § 1295(a)(4)(A).

DISCUSSION

Obviousness "is a question of law based on underlying findings of fact." *In re Gartside*, 203 F.3d 1305, 1316 (Fed. Cir. 2000) (citation omitted). We review the Board's factual findings for substantial evidence and its legal conclusions *de novo*. *Rambus Inc. v. Rea*, 731 F.3d 1248, 1251–52 (Fed. Cir. 2013).

I. TCP Data Segmentation

Substantial evidence supports the Board's finding that a skilled artisan would have motivated to have Erickson's scripts on its network interface implement TCP data segmentation as described in Tanenbaum with a reasonable expectation of success.

Just as Alacritech's patents' stated goal is to offload certain network processing tasks to a network interface card, Erickson likewise discloses how a network interface can be programmed using a script to allow user applications to transmit or receive network communications directly through the network interface, thereby "avoiding the overhead processing which would be incurred if the

operating system were used to service" such communications. *See* Erickson at col. 4 ll. 18–33, col. 4 l. 53–col. 5 l. 5, col. 5 ll. 15–22. Such scripts can be used "to define a particular set of instructions to be performed based upon the protocol type" of the communications. *Id.* at col. 5 ll. 41–47. As the Board recognized, Erickson's scripts handle various protocol types, including TCP and the User Datagram Protocol (UDP). *Id.* at col. 5 ll. 47–49; J.A. 147. Although Erickson does not provide a detailed example of a script for TCP, Erickson explains that the TCP protocol is "well-known within the art" and "may be found in many references," including an earlier edition of Tanenbaum expressly "incorporated by reference." *Id.* at col. 4 ll. 38–43.

Substantial evidence supports the Board's finding that it would have been obvious for a skilled artisan reading Erickson at the time of the invention to look to the then-current edition of Tanenbaum to determine what instructions Erickson's script should perform to handle TCP communications on Erickson's network interface. J.A. 147 ("although Erickson's exemplary context is UDP, it also discloses the use of TCP, refers readers to the 1981 edition of Tanenbaum, and discloses use of TCP/IP scripts"). Since Tanenbaum teaches that the TCP protocol segments data into separate packets for transmission, we see no error in the Board's conclusion that it would have been obvious for Erickson's network interface to implement such data segmentation using Erickson's TCP scripts, and substantial evidence supports the Board's finding that a skilled artisan would have been motivated to do so.[4]

---

[4] Specific to the '241 patent, Alacritech also argues that the Board failed to make any finding that a skilled artisan would be motivated to implement Tanenbaum's TCP processing on a network interface using Erickson's scripts. We disagree. As the Board explained, it was persuaded by petitioner's argument that "Tanenbaum *expressly discloses*

Alacritech argues that neither Erickson nor Tanenbaum discloses the use of a network interface to divide data into segments and the Board improperly relied on a skilled artisan's ordinary creativity to supply a missing limitation. But the Board did not rely on a broad assertion of "ordinary creativity." As the Board noted, Alacritech's arguments attack the references individually and not for what the combination teaches a skilled artisan. *See Soft Gel Techs., Inc. v. Jarrow Formulas, Inc.*, 864 F.3d 1334, 1341 (Fed. Cir. 2017); *In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986) The Board correctly reasoned that it would have been obvious to have Erickson's TCP scripts, whose purpose is to handle TCP processing tasks on behalf of the CPU, implement data segmentation, as Tanenbaum explains such segmentation is *required* by TCP. J.A. 98; J.A 147. Thus, substantial evidence supports the Board's finding that the combination of Erickson and Tanenbaum discloses the use of a secondary processor on a network interface to divide data into segments, and Alacritech's argument that this limitation is missing from the individual references is unpersuasive.

Substantial evidence also supports the Board's finding that Tanenbaum does not teach away from implementing TCP processing on a network interface. Although Tanenbaum states that, "[u]sually, the best approach is to make the protocols simple and have the main CPU do the work," J.A. 2959, Tanenbaum expressly discloses that processing of transport protocols (e.g., TCP) can be performed on a

---

that transport layer protocol [e.g., TCP] processing may be offloaded to a network interface card," i.e., a skilled artisan would be motivated by Tanenbaum's express teaching. J.A. 76–77 (emphasis added). The motivation is further reinforced by the fact that Erickson cites an earlier version of Tanenbaum to describe the TCP protocol. Erickson at col. 4 ll. 38–43.

network interface card. *Id.* at 2868 (explaining that the "transport entity" that handles the transport protocol can be "in the operating system kernel . . . or on the network interface card"). As the Board explained, "Tanenbaum actually discloses only that the system may not be *optimal* if a less 'expensive' CPU is selected and the 'slow CPU' 'do[es] the critical work." J.A. 157 (emphasis added). Alacritech also argues that a paper published by Dr. Horst supports its view that offloading TCP functions to a second processor was prohibitively expensive. But the Board was similarly unpersuaded because, as the petitioner explained, this paper also confirmed that "conventional wisdom" was that TCP was offloaded to special purpose network interface cards. J.A. 109–110; J.A. 175–76; *see also* J.A. 10239 ("Conventional wisdom says that IP storage is impractical without special purpose NICs to accelerate the TCP/IP protocol stack."). We note that the data segmentation claims do not require an optimal solution, but simply require "dividing, by the interface device, the data into segments." *See* '072 patent at claim 1. Considering Tanenbaum's express disclosure of offloading TCP to the network interface in light of Erickson, which likewise discloses that network protocol processing such as TCP can be performed on a network interface, substantial evidence supports the Board's finding that Tanenbaum's preference for processing on the "main CPU" did not teach away from the combination. J.A. 97–98; J.A 156–57.

II. TCP State Information

We also see no error in the Board's conclusion that updating TCP state information with a network interface would have been obvious over Erickson and Tanenbaum. Alacritech again complains that this limitation is missing from Erickson and Tanenbaum, but we find this attack on the individual references unpersuasive in view of what the references teach as a whole.

The Board found that Erickson teaches "fast-path processing," in which user applications can transmit data through a network interface while "avoiding the overhead" of normal processing. J.A. 7–8, 17. As explained above, Erickson's express reference to the earlier edition of Tanenbaum would lead a skilled artisan reading Erickson to Tanenbaum's disclosure for details on the TCP protocol. In turn, Tanenbaum describes how the process for receiving TCP packets can be simplified after an initial connection is established. If certain conditions are met, a received packet can, like in Erickson, undergo "fast path," as opposed to normal TCP processing. J.A. 2955. As Tanenbaum explains, this "fast path updates the connection record and copies the [received] data to the user." *Id.* Petitioner's expert, Dr. Horst, testified that Tanenbaum's "connection record" was "used to maintain the TCP state." J.A. 2319 (explaining that Tanenbaum discloses that "the local TCP entity creates a connection record" and "the state is per connection and recorded in the connection record"); *see also* J.A. 2919. Alacritech does not appear to dispute that Tanenbaum discloses updating TCP state as part of maintaining a "connection record." Thus, there is substantial evidence for the Board's finding that Tanenbaum's connection record was "used to maintain the TCP state," and the Board properly agreed with the petitioner that it would have been obvious to incorporate Tanenbaum's teachings on fast path into Erickson's fast path, resulting in a script for a network interface that updates TCP state information when receiving network packets. J.A. 23–24.

Alacritech urges that Erickson is limited to its disclosure of an exemplary script for transmitting packets, and thus its teachings cannot extend to receiving packets. We disagree. Contrary to Alacritech's suggestion, a skilled artisan would not read Erickson with a myopic focus on a single example for transmitting packets but would instead view the reference's teachings as a whole. As the petitioner explained, and the Board agreed, Erickson discloses that

its scripts can be used to "transmit *and receive information*" through the network interface. Erickson at col. 5 ll. 6–14 (emphasis added); J.A. 24. Substantial evidence supports the Board's finding regarding the scope of Erickson's disclosures.

Alacritech also argues that the Board failed to make any finding on motivation to combine. We disagree. The Board was persuaded that a skilled artisan "would have been motivated to implement 'fast path' TCP/IP on Erickson's [network interface] in view of Tanenbaum with a high expectation of success," particularly because Erickson expressly discloses that TCP scripts can be implemented for the network interface and in view of "TCP's popularity among a finite number [of] networking protocols." J.A. 11–12, 16–17. Because the Board also found that updating TCP state information is part and parcel of Tanenbaum's "fast path" TCP processing, the Board's findings with respect to motivation to combine on "fast path" TCP processing apply equally to updating TCP state information. J.A. 23–24.

CONCLUSION

We have considered Alacritech's remaining arguments and find them unpersuasive. For the reasons stated above, we *affirm* the Board's conclusion that the claims at issue are unpatentable.

AFFIRMED