

NOTE: This disposition is nonprecedential.

**United States Court of Appeals
for the Federal Circuit**

ALACRITECH, INC.,
Appellant

v.

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

UNITED STATES,
Intervenor

2019-1464

Appeal from the United States Patent and Trademark Office, Patent Trial and Appeal Board in Nos. IPR2017-01393, IPR2017-01714, IPR2018-00374.

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.

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, 6, 9, 12, and 15 of U.S. Patent No. 9,055,104 invalid as obvious.¹ We affirm.

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patent judges on the Board under the Appointments Clause of the Constitution, but this challenge has since been withdrawn and waived.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. 70.

BACKGROUND

The '104 patent relates to network communications, and in particular to the use of a network interface device to offload network processing tasks from the central processing unit (CPU) of a host computer. '104 patent at Abstract. To transmit data to the network, the computer supplies a command along with the data to the network interface, which in turn processes and transmits that data according to the Transmission Control Protocol (TCP) before indicating to the computer that the data has been transmitted. *Id.* at col. 3 ll. 37–47. TCP also requires the recipient of the data to respond with acknowledgments (ACKs) indicating to the transmitting device—here, the network interface—which data has been successfully received. *Id.* at col. 2 ll. 10–15. The claimed invention specifies that the network interface indicates to the transmitting computer that the data has been transmitted “prior to receiving” the ACK for all of the transmitted data. Claim 1 is representative:

1. A method for communication involving a computer, a network, and a network interface device of the computer, the network interface device being coupled to the network, the method comprising:

receiving, by the network interface device from the computer, a command to transmit application data from the computer to the network;

sending, by the network interface device to the network, data corresponding to the command, including prepending a transport layer header to at least some of the data;

sending, by the network interface device to the computer, a response to the command indicating that the data has been sent from the network interface device to the network, prior to receiving, by the network interface device from the network, an

acknowledgement (ACK) that all the data corresponding to the command has been received; and

maintaining, by the network interface device, a Transport Control Protocol (TCP) connection that the command, the data and the ACK correspond to.

Id. at claim 1 (emphasis added).

The Board found all of the challenged claims unpatentable over U.S. Patent No. 5,937,169 (Connery) in view of the knowledge of a skilled artisan and, in the alternative, over Connery in view of PCT Patent Publication No. WO 00/13091. Relevant to this appeal, the Board found that Connery discloses the limitation of “sending, by the network interface device to the computer, a response to the command indicating that the data has been sent from the network interface device to the network,” and that Connery’s network interface sends this response “prior to receiving . . . an acknowledgment (ACK) that all the data corresponding to the command has been received.” J.A. 19–25; ’104 patent at claim 1.

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 findings regarding the scope and content of the prior art for substantial evidence. *Rambus Inc. v. Rea*, 731 F.3d 1248, 1251–52 (Fed. Cir. 2013).

The sole dispute in this appeal relates to what Connery discloses, and specifically, the relative timing between a particular “interrupt” message sent from Connery’s network interface to the computer’s central processing unit (CPU) and the receipt of an ACK sent from a computer that received the transmitted data.

Like the '104 patent, Connery discloses that network protocol processing can be offloaded from the CPU to a network interface. Connery at col. 1 ll. 7–11. As Connery explains, higher layer network protocols such as TCP were typically “handled by software drivers” and “a protocol stack executed in the host processor,” i.e., CPU. *Id.* at col. 1 ll. 20–21, 31–33. To transmit data according to TCP, the CPU would first package that data into “appropriately sized segments.” *Id.* at col. 1 ll. 37–41. Separately, the CPU would also monitor acknowledgment (ACK) messages from the recipient to ensure, for reliability purposes, that “the amount of data sent onto the network” “does not exceed” the advertised receiving capacity of the recipient. *Id.* at col. 1 l. 56–col. 2 l. 13. To reduce the load on the CPU, Connery explains that segmentation of data for transmission can be offloaded from the CPU to the network interface. *Id.* at col. 7 ll. 47–49, 60–64. With the exception of data segmentation, Connery’s CPU continues to handle the remaining aspects of the TCP protocol. *Id.* at col. 5 ll. 51–53, col. 6 ll. 15–18.

Connery’s network interface accepts a command from the CPU to transmit a “large datagram” and divides that datagram into individually transmitted segments according to the TCP network protocol. *Id.* at col. 2 ll. 46–57. Once the network interface transmits the data segments of the large datagram, Connery discloses that the network interface can send a single interrupt to the CPU. *Id.* at col. 7 ll. 60–63. The parties dispute whether this interrupt sent by the network interface occurs before, or after, the network interface receives an ACK indicating that all of the transmitted data was successfully received by a destination computer.

The Board agreed with the petitioner that a skilled artisan would understand Connery’s single interrupt per large datagram as indicating that “all generated packets from the large datagram have been sent from the network interface to the network regardless of whether any

corresponding ACK[s] have been received.” J.A. 23. The Board further found that Connery’s interrupt would be generated by the network interface prior to receiving an ACK from the recipient for all of the transmitted data. Substantial evidence supports the Board’s fact findings.

The Board reasonably found that the interrupt generated by Connery’s network interface is strictly associated with the task given to the network interface—transmitting data in segments—and not other tasks, such as ACK processing, that the host computer has not offloaded to the network interface. As Intel’s expert, Dr. Horst, explained, such an interrupt to indicate that the network interface is done with segmenting and transmitting data “is needed because otherwise the host would not know when the [network interface] hardware is free to accept the next large packet to be sent.” J.A. 1670. In contrast, the Board found that ACK processing is “unaffected by Connery’s alleged improvements to segmenting and transmitting data” on the network interface. J.A. 23. Thus, “an ordinarily skilled artisan would have presumed ACKs are processed in the normal processing of a TCP/IP protocol stack operable in the host computer.” *Id.* In other words, while Connery contemplates offloading the task of transmitting data in segments from the CPU to its network interface, which in turn sends an interrupt back to the host when that task is complete, ACK processing is not affected; the ACK, as a separate message, continues to be sent from the recipient and handled by the CPU in a conventional manner. Because ACK processing has not been offloaded from Connery’s CPU to its network interface, the Board fairly concluded that the “interrupt” generated by the network interface indicates that the interface has completed its data transmission assignment rather than referring to the ACK. Thus, like the Board, we are not persuaded by Alacritech’s argument that Connery’s interrupt must include the ACK or depend on receipt of the ACK.

After determining that Connery’s “interrupt for transmit complete” is sent by the network interface to the CPU independently of the receipt of any ACK, the Board considered the relative timing between the transmit complete interrupt and the return of ACKs for all of the transmitted data. J.A. 24. We see no error in the Board’s reasoning that “an interrupt for transmit complete would precede” the ACK indicating that all data segments of the datagram have been transmitted “because there is substantial latency in receipt of the ACK as compared to” the interrupt. *Id.* After the network interface “transmit[s] the last packet of the datagram,” the transmit complete interrupt would be “sent immediately” and “virtually instantaneous” between the network interface and the CPU, whereas there would be “substantial latency,” i.e., delay, from the transmission of the last packet of the datagram over the network, the processing of that packet at the receiving computer, and the return transmission of an ACK for that packet from the receiving computer to the transmitting network interface. *Id.*

Alacritech alleges that, under the Board’s reading, Connery’s network interface must also generate a second interrupt “to let the CPU know that the ACKs have been received and the data transfer is complete,” rendering the initial transmit complete interrupt “meaningless.” Appellant’s Br. at 20–22. According to Alacritech, the CPU must wait for this alleged second interrupt indicating receipt of the ACKs for the first batch of data before “sending the next batch of data to the [network interface] for transmission.” *Id.* We note that Alacritech did not make this “second interrupt” argument before the Board, and arguments not made below are generally waived. *In re NuVasive, Inc.*, 842 F.3d 1376, 1380 (Fed. Cir. 2016) (“a party waives an argument that it ‘failed to present to the [PTAB]’ because it deprives the court of ‘the benefit of the [PTAB]’s informed judgment”) (citing *In re Watts*, 354 F.3d 1362, 1367–68 (Fed. Cir. 2004)). Although Alacritech points out that it

argued to the Board that a skilled artisan would not depart from the allegedly “conventional” practice in which “the network interface device waits for an ACK for all the transmitted data before it sends a response indicating the command has been completed to the local host,” that says nothing about a second interrupt in addition to Connery’s or that Connery’s CPU must wait for a second interrupt before supplying additional data to its network interface for transmission. Appellant’s Reply Br. at 1–2 (citing J.A. 444). To the extent that Alacritech argued that Connery’s interrupt is dependent on the receipt of an ACK, we explained above that substantial evidence supports the Board’s conclusion that Connery’s interrupt is indicative of transmit completion rather than ACK receipt.

In any event, nothing in Connery or the record suggests that Connery’s CPU must wait for ACK confirmation before supplying additional data to the network interface for transmission. Alacritech does not contend that the TCP protocol itself imposes such a requirement on the interaction between the CPU and the network interface. Appellant’s Reply Br. at 8–9 (acknowledging that TCP’s flow control mechanism, which includes ACKs, “does not specify the communications between the sending computer’s CPU and a [network interface card]”). As both parties agree, the TCP protocol controls the flow of data between transmitting and receiving devices based on a “window” advertised by the receiving device. The receiving window defines how much data the transmitting device can send to the receiving device without waiting for ACKs from the receiving device. Connery itself explains that the sending computer can continue to transmit data without waiting for any ACKs so long as the “sender controls the amount of data sent onto the network so that it does not exceed the size of the advertised window of the destination.” Connery at col. 2 ll. 3–13. Connery further confirms that, consistent with TCP’s constraints on transmitting data, the CPU can “send[] a plurality of datagrams to the network interface

for the same session,” and the network interface will “concatenate[],” i.e., link, “data from the current datagram” that is sequentially aligned “with data from the following datagram.” *Id.* at col. 4 ll. 34–41. As Intel explains, such concatenation between datagrams would be impossible if, as Alacritech urges, the following datagram was not made available to the network interface until after the current datagram is transmitted and an ACK received.

In sum, the Board’s decision finding that Connery’s interrupt represents the network interface’s completion of transmission of a large datagram and occurs prior to the receipt of an ACK for all of the data in that large datagram is supported by substantial evidence.

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