Intel Corp. v. Alacritech, Inc.

IPR2018-00226, -00234, -00401

March 4, 2019

Demonstratives: Table of Contents

Demonstratives: Table of Contents

(3) 699 Patent (IPR2018-00401)

Demonstratives: Table of Contents

U.S. Patent No. 8,805,948
(948 Patent)

IPR2018-00234 (Intel)
IPR2018-00403 (Cavium)
IPR2018-01307 (Dell)

*All citations herein are to the IPR2018-00234 case unless otherwise noted.

948 Patent: Instituted Grounds

- **Thia**, **Tannenbaum96**, and **Stevens2**: claims **1**, 3, 6-8, **17**, 19, 21, and 22.

948 Patent: Disputes

1. A POSA would have been motivated to combine Thia and Tanenbaum96 (and Stevens2)

2. The prior art combinations disclose the limitations of the challenged claims of the 948 Patent

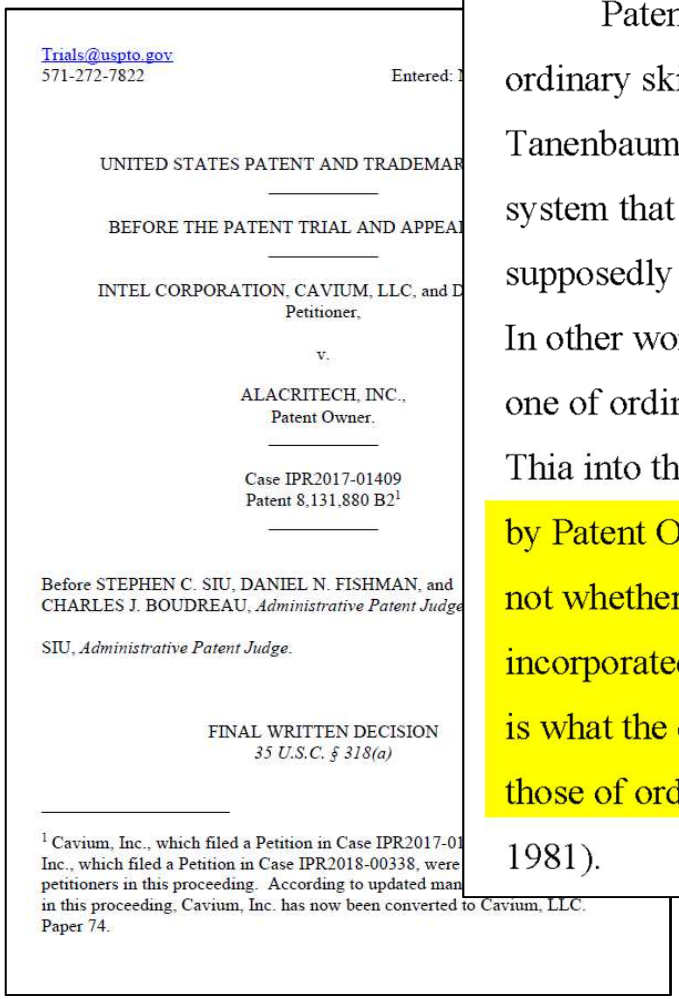
The Board Has Rejected Many of PO's Arguments

- This Petition involves overlapping prior art and arguments as in prior related IPRs, including IPRs on the 880 Patent (IPR2018-01409; IPR2018-01410)
- Board has previously rejected PO's arguments
  - -01409 FWD at 11-14: finding it would have been obvious to combine Thia and Tanenbaum96
  - -1409 FWD at 10-11: finding that Thia and Tanenbaum96 teach storing data on the host without TCP headers

948 Patent: Disputes

1. A POSA would have been motivated to combine Thia and Tanenbaum96 (and Stevens2)
   a. **Tanenbaum96 does not teach away from the combination (Board previously sided with Petitioner)**
   b. The trend towards TCP/IP in the 1990s would motivate combining Thia's bypass architecture with TCP/IP (Board previously sided with Petitioner)
   c. A POSA would have understood that Thia's teachings are applicable to TCP/IP (Board previously sided with Petitioner)
   d. It would have been obvious to combine Stevens2 with Thia and Tanenbaum96

The Board rejected PO's argument that Tanenbaum96 teaches away



Trials@uspto.gov
571-272-7822 Entered: 1

UNITED STATES PATENT AND TRADEMAR

BEFORE THE PATENT TRIAL AND APPEAL

INTEL CORPORATION, CAVIUM, LLC, and D
Petitioner,

v.

ALACRITECH, INC.,
Patent Owner.

Case IPR2017-01409
Patent 8,131,880 B2[1]

Before STEPHEN C. SIU, DANIEL N. FISHMAN, and CHARLES J. BOUDREAU, *Administrative Patent Judge*

SIU, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a)

[1] Cavium, Inc., which filed a Petition in Case IPR2017-01 Inc., which filed a Petition in Case IPR2018-00338, were petitioners in this proceeding. According to updated man in this proceeding, Cavium, Inc. has now been converted to Cavium, LLC. Paper 74.

Patent Owner argues that it would not have been obvious to one of ordinary skill in the art to have combined the teachings of Thia and Tanenbaum because, according to Patent Owner, Tanenbaum discloses a system that "does not introduce a separate processor" but that Thia supposedly discloses a system that has a separate processor. PO Resp. 49. In other words, Patent Owner argues that it would not have been obvious to one of ordinary skill in the art to have bodily incorporated the processor of Thia into the system of Tanenbaum (or vice versa). We are not persuaded by Patent Owner's argument at least because "[t]he test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. . . . Rather, the test is what the combined teachings of those references would have suggested to those of ordinary skill in the art." *In re Keller*, 642 F.2d 413, 425 (CCPA 1981).

IPR2017-01409 Paper 79 (FWD) at 12;
see also Paper 35 (Reply) at 4-5.

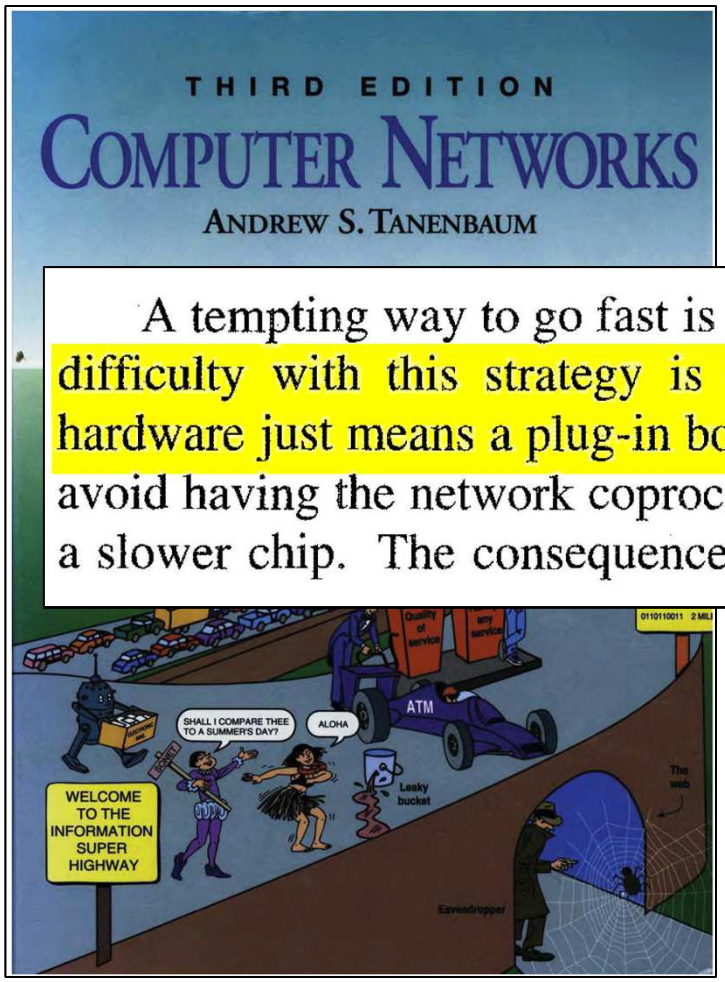
Tanenbaum96 does not teach away from a combination with Thia

Instead, it describes design preferences and tradeoffs

A tempting way to go fast is to build fast network interfaces in hardware. The difficulty with this strategy is that unless the protocol is exceedingly simple, hardware just means a plug-in board with a second CPU and its own program. To avoid having the network coprocessor be as expensive as the main CPU, it is often a slower chip. The consequence of this design is that much of the time the main (fast) CPU is idle waiting for the second (slow) CPU to do the critical work. It is a myth to think that the main CPU has other work to do while waiting. Furthermore, when two general-purpose CPUs communicate, race conditions can occur, so elaborate protocols are needed between the two processors to synchronize them correctly. Usually, the best approach is to make the protocols simple and have the main CPU do the work.

Ex. 1006.588-.589 (Tanenbaum96);
see also Paper 35 (Reply) at 4; Ex. 1399.027-.029 (Horst Reply Decl.).

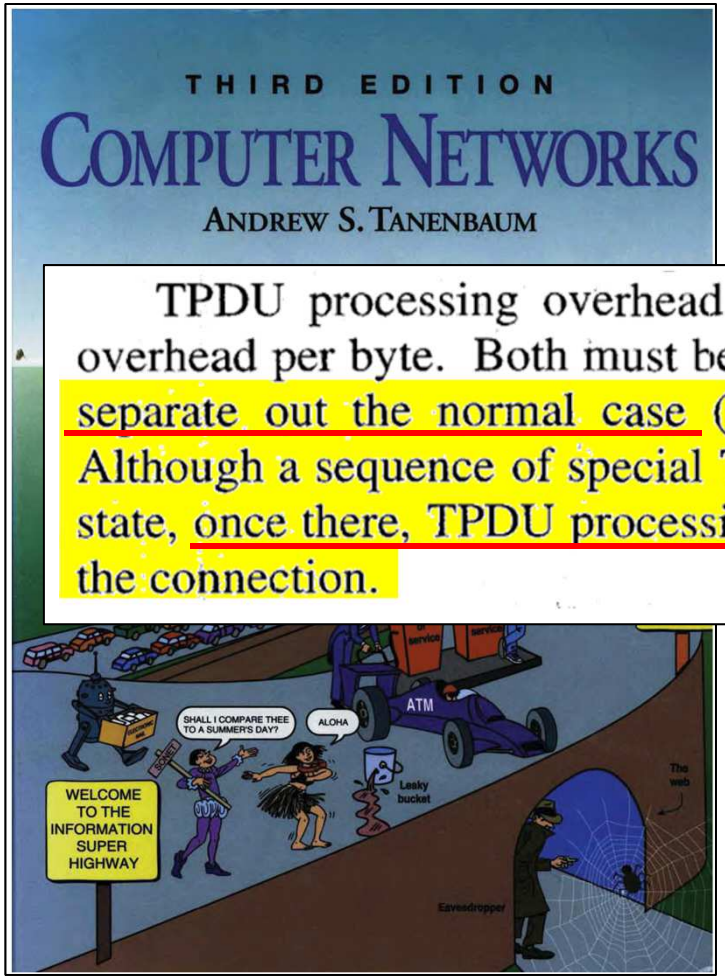
Tanenbaum96 does not discourage offloading simple protocols



A tempting way to go fast is to build fast network interfaces in hardware. The difficulty with this strategy is that unless the protocol is exceedingly simple, hardware just means a plug-in board with a second CPU and its own program. To avoid having the network coprocessor be as expensive as the main CPU, it is often a slower chip. The consequence of this design is that much of the time the main

Ex. 1006.588 (Tanenbaum96);
see also Paper 35 (Reply) at 4, 7;
Ex.1399.028-.029 (Horst Reply Decl.).

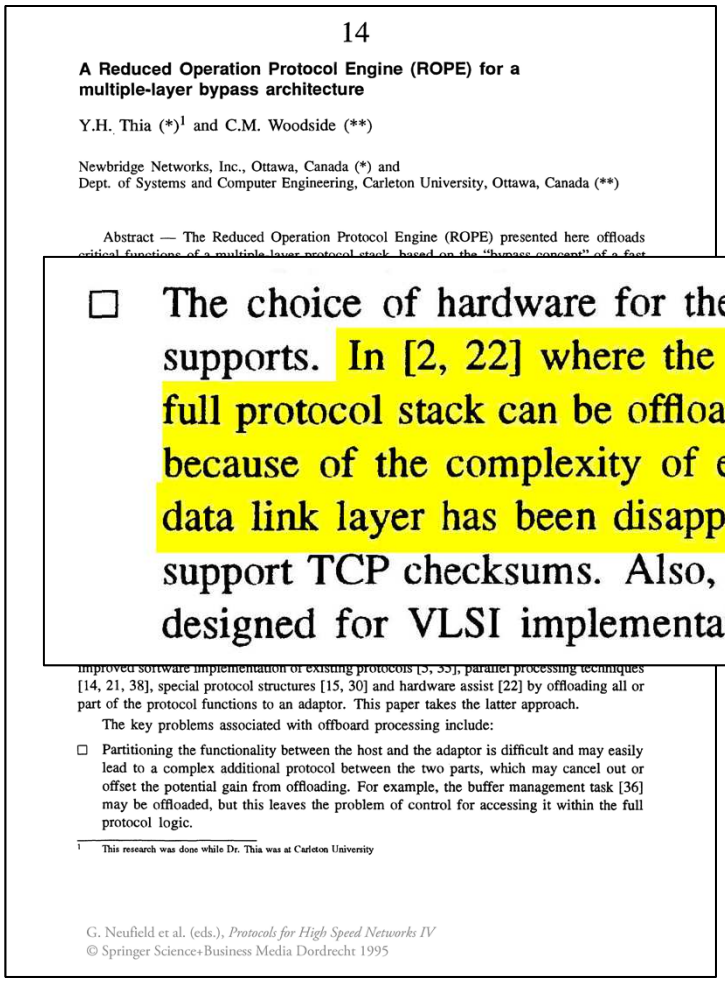
Tanenbaum96: Transport processing is "straightforward" in the "normal case"



TPDU processing overhead has two components: overhead per TPDU and overhead per byte. Both must be attacked. The key to fast TPDU processing is to separate out the normal case (one-way data transfer) and handle it specially. Although a sequence of special TPDUs are needed to get into the *ESTABLISHED* state, once there, TPDU processing is straightforward until one side starts to close the connection.

Ex. 1006.583 (Tanenbaum96);
see also Paper 35 (Reply) at 4.

Thia also recognizes the difficulty of offloading a complex protocol stack



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple layer protocol stack based on the "bypass concept" of a fast

improved software implementation of existing protocols [5, 33], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

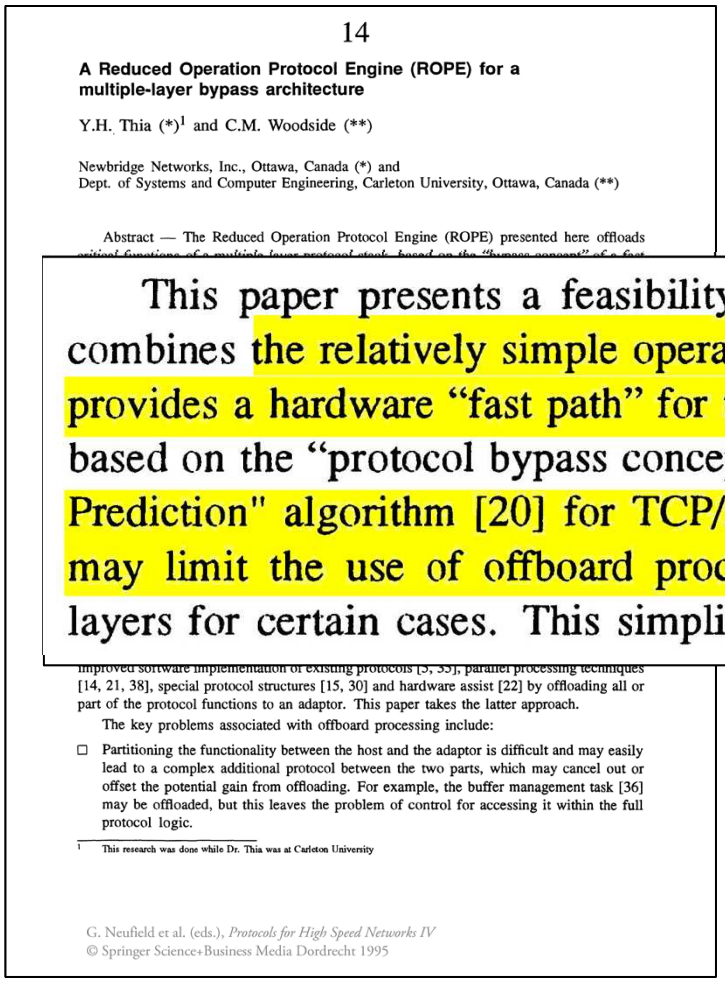
[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

- The choice of hardware for the adaptor depends on the complexity of the functions it supports. In [2, 22] where the transport protocol layer is offloaded or in [7] where the full protocol stack can be offloaded, general purpose microprocessors are used. Probably because of the complexity of existing protocols, VLSI [24] implementation above the data link layer has been disappointing so far. In [8], dedicated VLSI chips are used to support TCP checksums. Also, some newer lightweight transport protocols are specially designed for VLSI implementation [1, 3].

Ex. 1015.002 (Thia);
see also Paper 35 (Reply) at 4.

Thia's solution: "Fast path" offload is based on header prediction



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads

This paper presents a feasibility study for a new approach to hardware assistance. It combines the relatively simple operations needed for data transfer across multiple layers and provides a hardware "fast path" for them, which will be efficient for bulk data transfer. It is based on the "protocol bypass concept" [37] which is a generalization of Jacobson's "Header Prediction" algorithm [20] for TCP/IP. Bypass solves the problems identified above, which may limit the use of offboard processing, by implementing an entire service through all layers for certain cases. This simplifies the interface between the host and the adaptor chip

improved software implementation of existing protocols [5, 33], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

Ex. 1015.002 (Thia);
see also Paper 35 (Reply) at 4.

Both disclose a bypass/fast-path based on TCP/IP header prediction

14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

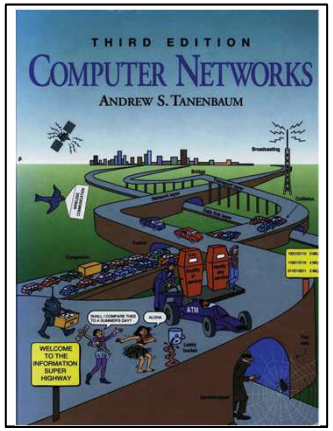
- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

This paper presents a feasibility study for a new approach to hardware assistance. It combines the relatively simple operations needed for data transfer across multiple layers and provides a hardware "fast path" for them, which will be efficient for bulk data transfer. It is based on the "protocol bypass concept" [37] which is a generalization of Jacobson's "Header Prediction" algorithm [20] for TCP/IP. Bypass solves the problems identified above, which may limit the use of offboard processing, by implementing an entire service through all layers for certain cases. This simplifies the interface between the host and the adaptor chip and minimizes their interaction, which is supported by an access test, some DMA processing and a simple command protocol. The chip design based on bypassing is called ROPE, for Reduced Operation Protocol Engine. The contribution of this paper is to define the host/chip interface and the chip operation, and to report on a VHDL-based feasibility study of the chip design. It appears to be feasible to support an end-system single-connection data rate approaching 1 Gbps.

Ex. 1015.002 (Thia);
see also Paper 35 (Reply) at 2.



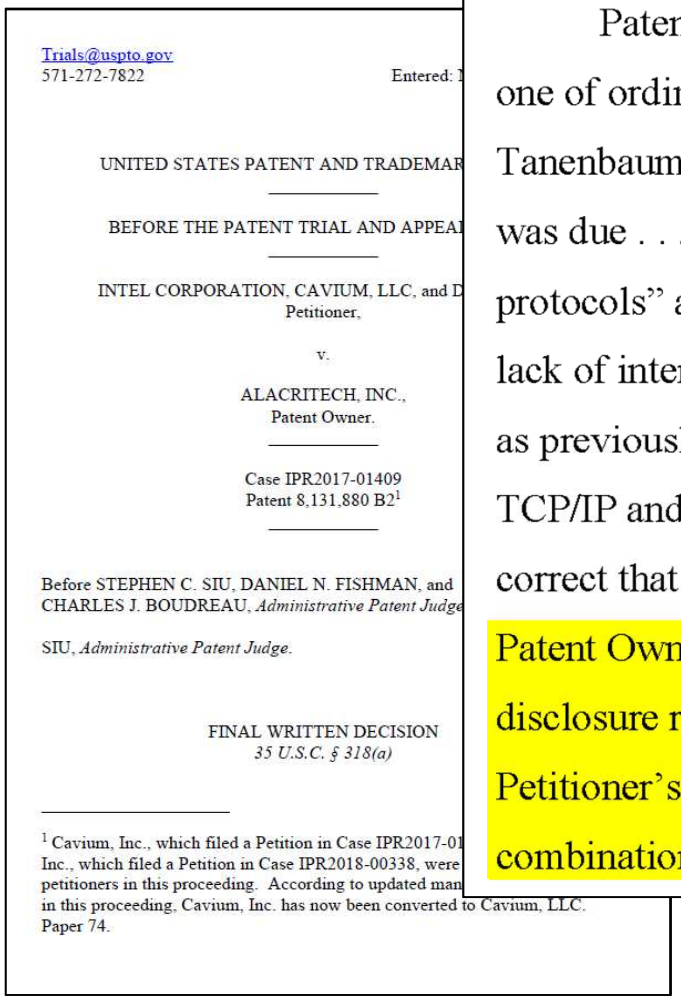
The fast path updates the connection record and copies the data to the user. While it is copying, it also computes the checksum, eliminating an extra pass over the data. If the checksum is correct, the connection record is updated and an acknowledgement is sent back. The general scheme of first making a quick check to see if the header is what is expected, and having a special procedure to handle that case, is called **header prediction**. Many TCP implementations use it. When this optimization and all the other ones discussed in this chapter are used together, it is possible to get TCP to run at 90 percent of the speed of a local memory-to-memory copy, assuming the network itself is fast enough.

Ex. 1006.585 (Tanenbaum96);
see also Paper 35 (Reply) at 2.

948 Patent: Disputes

1. A POSA would have been motivated to combine Thia and Tanenbaum96 (and Stevens2)
   a. Tanenbaum96 does not teach away from the combination (Board previously sided with Petitioner)
   b. The trend towards TCP/IP in the 1990s would motivate combining Thia's bypass architecture with TCP/IP (Board previously sided with Petitioner)
   c. A POSA would have understood that Thia's teachings are applicable to TCP/IP (Board previously sided with Petitioner)
   d. It would have been obvious to combine Stevens2 with Thia and Tanenbaum96

The Board rejected PO's "lack of interest in OSI" argument



Trials@uspto.gov
571-272-7822 Entered: 1

UNITED STATES PATENT AND TRADEMAR

BEFORE THE PATENT TRIAL AND APPEAL

INTEL CORPORATION, CAVIUM, LLC, and D
Petitioner,

v.

ALACRITECH, INC.,
Patent Owner.

Case IPR2017-01409
Patent 8,131,880 B2[1]

Before STEPHEN C. SIU, DANIEL N. FISHMAN, and CHARLES J. BOUDREAU, *Administrative Patent Judge*

SIU, *Administrative Patent Judge*.

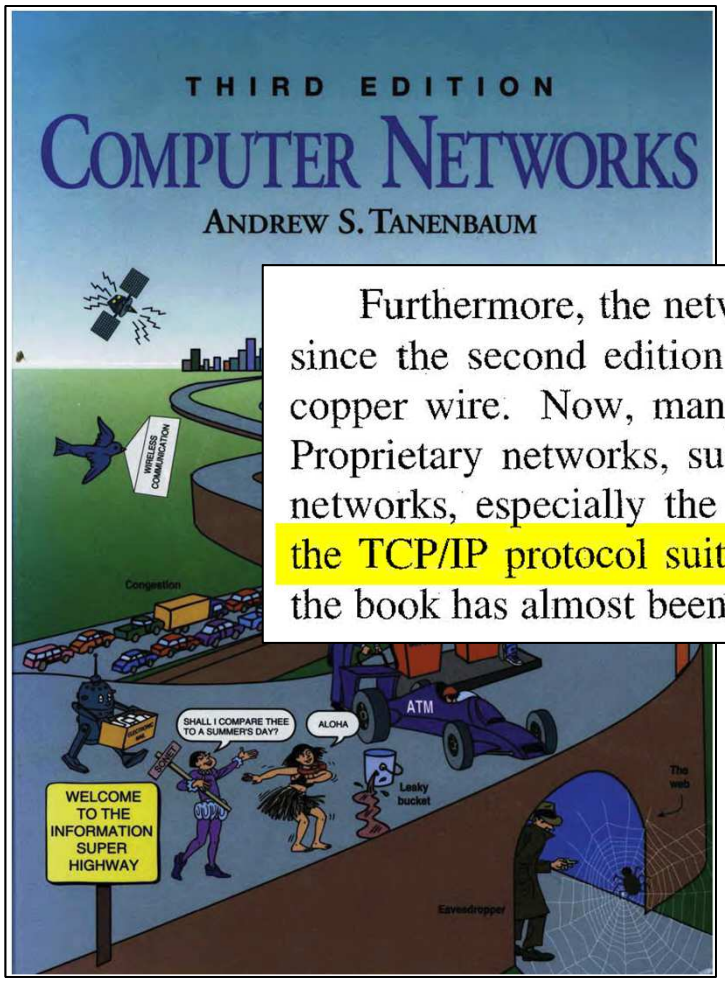
FINAL WRITTEN DECISION
35 U.S.C. § 318(a)

[1] Cavium, Inc., which filed a Petition in Case IPR2017-01 Inc., which filed a Petition in Case IPR2018-00338, were petitioners in this proceeding. According to updated man in this proceeding, Cavium, Inc. has now been converted to Cavium, LLC. Paper 74.

Patent Owner further argues that it would not have been obvious to one of ordinary skill in the art to have combined the teachings of Thia and Tanenbaum because "Tanenbaum explains that the lack of interest in OSI was due . . . to 'the enormous complexity of the [OSI] model and the protocols" and that, according to Patent Owner, there was an "undisputed lack of interest in OSI in the relevant timeframe." PO Resp. 51. However, as previously discussed, Petitioner relies on Tanenbaum for disclosing TCP/IP and not OSI. Even assuming Patent Owner's contention to be correct that Tanenbaum supposedly discloses a "lack of interest in OSI," Patent Owner does not assert or demonstrate persuasively that this presumed disclosure regarding an alleged "lack of interest in OSI" sufficiently refutes Petitioner's showing of obviousness of the disputed claims over the combination of Thia and Tanenbaum.

IPR2017-01409 Paper 79 (FWD) at 13;
see also Paper 35 (Reply) at 5.

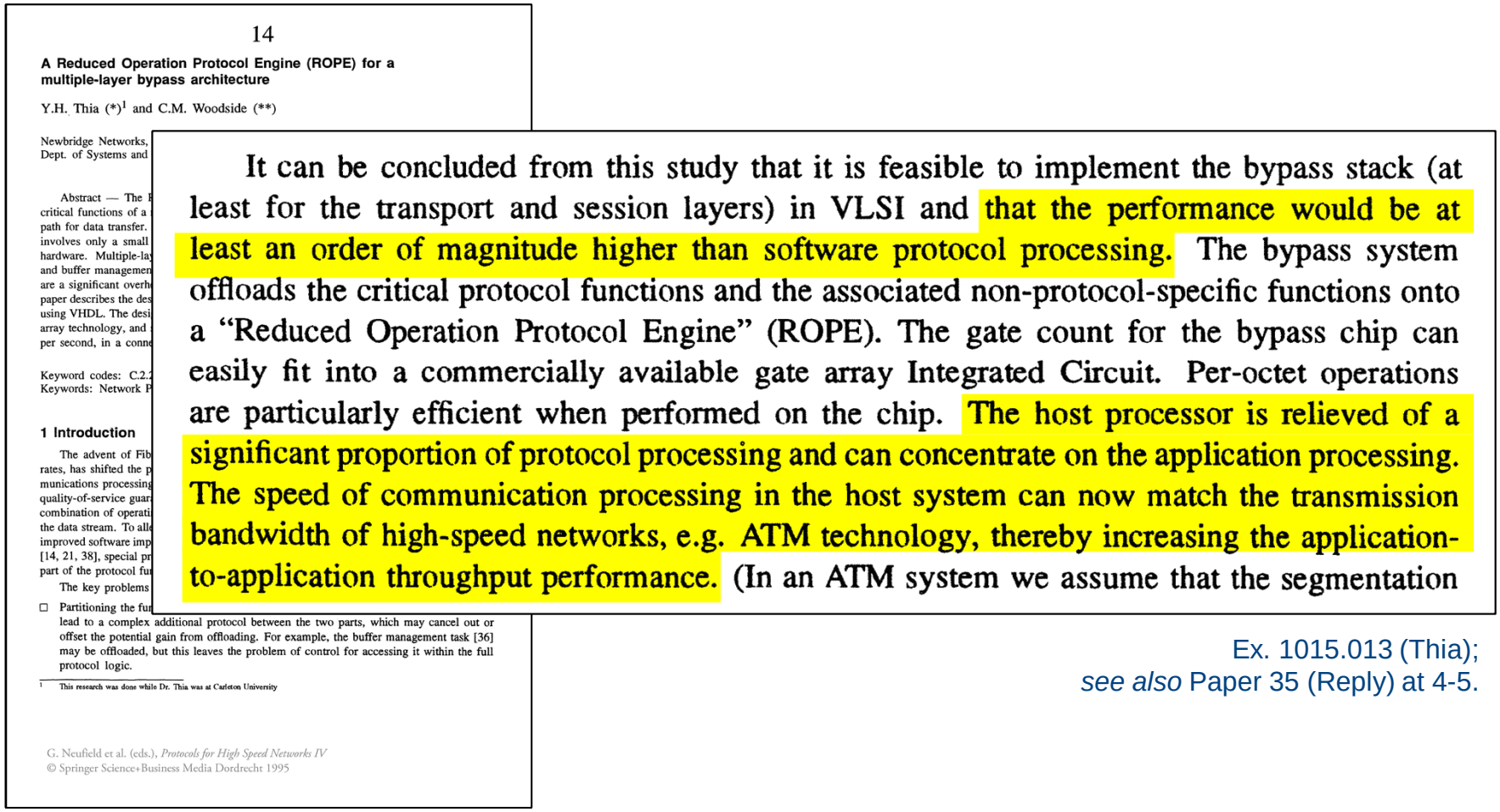
By 1996 OSI protocol use vanished and TCP/IP became dominant



Furthermore, the networking hardware and software have completely changed since the second edition appeared. In 1988, nearly all networks were based on copper wire. Now, many are based on fiber optics or wireless communication. Proprietary networks, such as SNA, have become far less important than public networks, especially the Internet. The OSI protocols have quietly vanished, and the TCP/IP protocol suite has become dominant. In fact, so much has changed, the book has almost been rewritten from scratch.

Ex. 1006.016 (Tanenbaum96);
see also Paper 2 (Petition) at 25, 57.

Thia's hardware offload provides advantages over software alone



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks,
Dept. of Systems and

Abstract — The
critical functions of a
path for data transfer.
involves only a small
hardware. Multiple-la
and buffer managemen
are a significant overh
paper describes the des
using VHDL. The desi
array technology, and
per second, in a conn

Keyword codes: C.2.
Keywords: Network

1 Introduction

The advent of Fib
rates, has shifted the p
munications processing
quality-of-service guar
combination of operati
the data stream. To all
improved software imp
[14, 21, 38], special pr
part of the protocol fu

The key problems

- Partitioning the fu
  lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

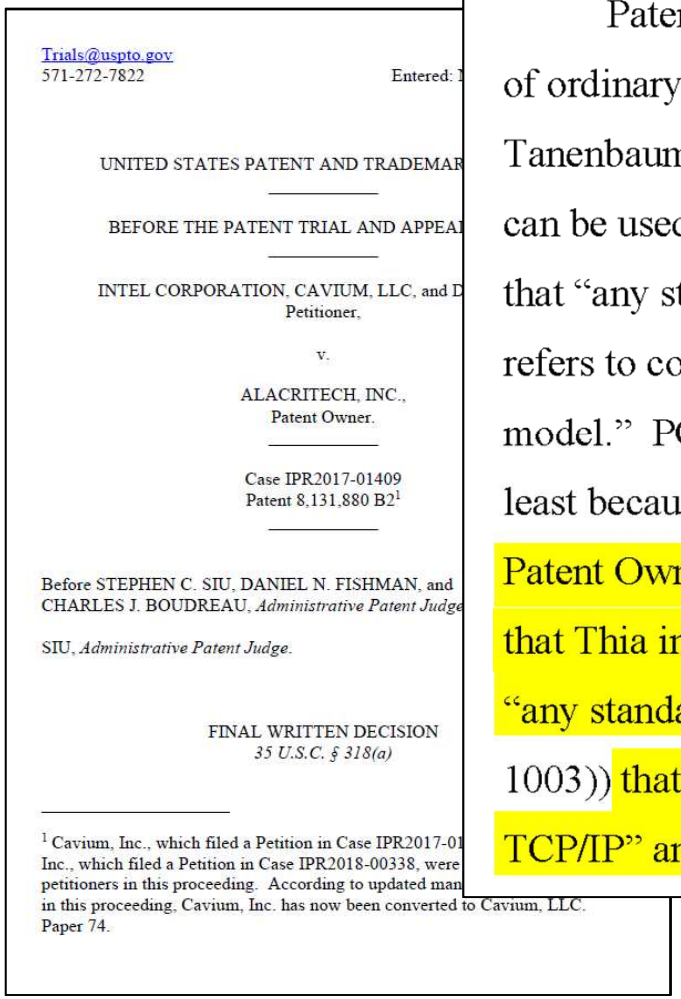
It can be concluded from this study that it is feasible to implement the bypass stack (at least for the transport and session layers) in VLSI and that the performance would be at least an order of magnitude higher than software protocol processing. The bypass system offloads the critical protocol functions and the associated non-protocol-specific functions onto a "Reduced Operation Protocol Engine" (ROPE). The gate count for the bypass chip can easily fit into a commercially available gate array Integrated Circuit. Per-octet operations are particularly efficient when performed on the chip. The host processor is relieved of a significant proportion of protocol processing and can concentrate on the application processing. The speed of communication processing in the host system can now match the transmission bandwidth of high-speed networks, e.g. ATM technology, thereby increasing the application-to-application throughput performance. (In an ATM system we assume that the segmentation

Ex. 1015.013 (Thia);
see also Paper 35 (Reply) at 4-5.

948 Patent: Disputes

1. A POSA would have been motivated to combine Thia and Tanenbaum96 (and Stevens2)
   a. Tanenbaum96 does not teach away from the combination (Board previously sided with Petitioner)
   b. The trend towards TCP/IP in the 1990s would motivate combining Thia's bypass architecture with TCP/IP (Board previously sided with Petitioner)
   c. A POSA would have understood that Thia's teachings are applicable to TCP/IP (Board previously sided with Petitioner)
   d. It would have been obvious to combine Stevens2 with Thia and Tanenbaum96

The Board rejected PO's "standard OSI protocol" argument



Trials@uspto.gov
571-272-7822 Entered:

UNITED STATES PATENT AND TRADEMAR

BEFORE THE PATENT TRIAL AND APPEA

INTEL CORPORATION, CAVIUM, LLC, and D
Petitioner,

v.

ALACRITECH, INC.,
Patent Owner.

Case IPR2017-01409
Patent 8,131,880 B2[1]

Before STEPHEN C. SIU, DANIEL N. FISHMAN, and CHARLES J. BOUDREAU, *Administrative Patent Judge*

SIU, *Administrative Patent Judge*.

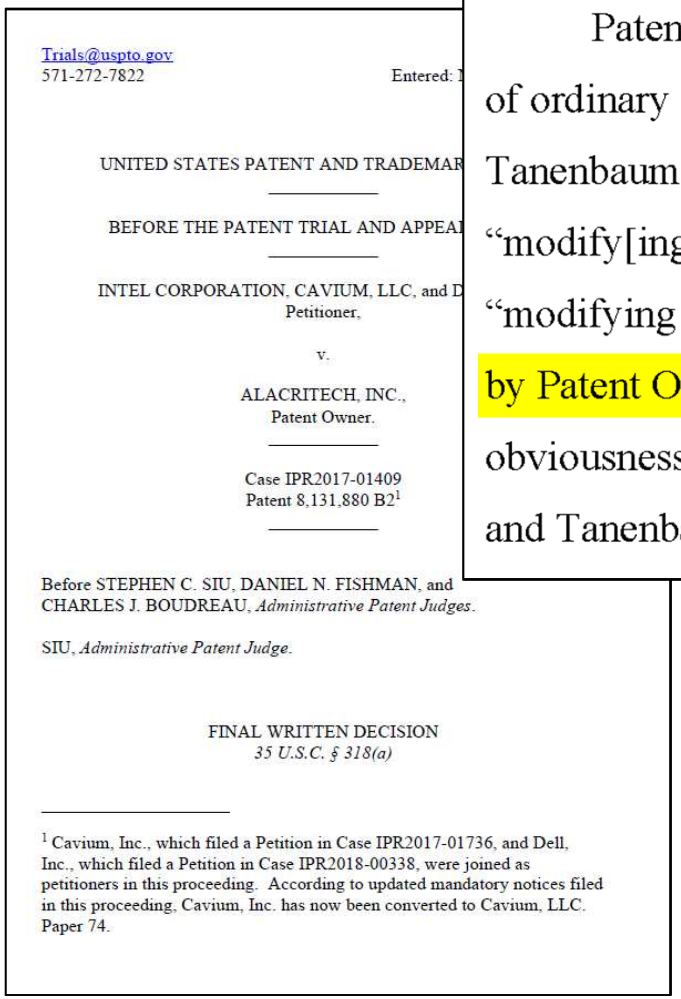
FINAL WRITTEN DECISION
35 U.S.C. § 318(a)

[1] Cavium, Inc., which filed a Petition in Case IPR2017-01 Inc., which filed a Petition in Case IPR2018-00338, were petitioners in this proceeding. According to updated man in this proceeding, Cavium, Inc. has now been converted to Cavium, LLC. Paper 74.

Patent Owner also argues that it would not have been obvious to one of ordinary skill in the art to have combined the teachings of Thia and Tanenbaum because Thia allegedly discloses that "its bypass architecture can be used with 'any standard protocol'" but supposedly intends to disclose that "any standard protocol" includes only "OSI protocols" because "Thia refers to concepts and features that are part of the OSI model, not the TCP/IP model." PO Resp. 51. We are not persuaded by Patent Owner's argument at least because Patent Owner does not provide sufficient evidence supporting Patent Owner's allegation that one of skill in the art would have understood that Thia intended to disclose "any OSI protocol" but inadvertently discloses "any standard protocol." We agree with Petitioner (Pet. 30–35 (citing Ex. 1003)) that "Thia's bypass stack is a generalization of the . . . algorithm for TCP/IP" and is not "confined to the OSI protocol." Pet. Reply 9. In

IPR2017-01409 Paper 79 (FWD) at 13;
see also Paper 35 (Reply) at 5.

The Board rejected PO's argument that Thia teaches away from using TCP/IP



Trials@uspto.gov
571-272-7822 Entered: 1

UNITED STATES PATENT AND TRADEMAR

BEFORE THE PATENT TRIAL AND APPEAL

INTEL CORPORATION, CAVIUM, LLC, and D
Petitioner,

v.

ALACRITECH, INC.,
Patent Owner.

Case IPR2017-01409
Patent 8,131,880 B2[1]

Before STEPHEN C. SIU, DANIEL N. FISHMAN, and CHARLES J. BOUDREAU, *Administrative Patent Judges*.

SIU, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a)

[1] Cavium, Inc., which filed a Petition in Case IPR2017-01736, and Dell, Inc., which filed a Petition in Case IPR2018-00338, were joined as petitioners in this proceeding. According to updated mandatory notices filed in this proceeding, Cavium, Inc. has now been converted to Cavium, LLC. Paper 74.

Patent Owner also argues that it would not have been obvious to one of ordinary skill in the art to have combined the teachings of Thia and Tanenbaum because Thia supposedly discloses "an easy migration path" by "modify[ing] existing ***OSI*** stack software" but supposedly fails to disclose "modifying ***TCP/IP*** stack software." PO Resp. 53. We are not persuaded by Patent Owner's argument at least because the Petitioner's showing of obviousness of the claimed invention is based on the combination of Thia and Tanenbaum and not based on Thia alone.

IPR2017-01409 Paper 79 (FWD) at 14;
see also Paper 35 (Reply) at 6-7.

Both disclose a bypass/fast-path based on TCP/IP header prediction

14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

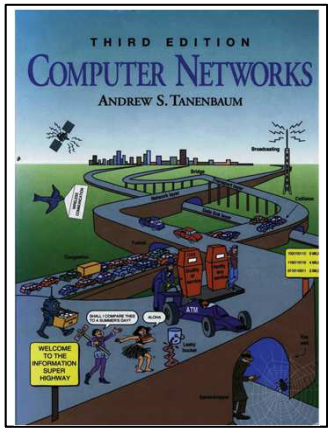
- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High-Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

This paper presents a feasibility study for a new approach to hardware assistance. It combines the relatively simple operations needed for data transfer across multiple layers and provides a hardware "fast path" for them, which will be efficient for bulk data transfer. It is based on the "protocol bypass concept" [37] which is a generalization of Jacobson's "Header Prediction" algorithm [20] for TCP/IP. Bypass solves the problems identified above, which may limit the use of offboard processing, by implementing an entire service through all layers for certain cases. This simplifies the interface between the host and the adaptor chip and minimizes their interaction, which is supported by an access test, some DMA processing and a simple command protocol. The chip design based on bypassing is called ROPE, for Reduced Operation Protocol Engine. The contribution of this paper is to define the host/chip interface and the chip operation, and to report on a VHDL-based feasibility study of the chip design. It appears to be feasible to support an end-system single-connection data rate approaching 1 Gbps.

Ex. 1015.002 (Thia); *see also* Paper 35 (Reply) at 2.



The fast path updates the connection record and copies the data to the user. While it is copying, it also computes the checksum, eliminating an extra pass over the data. If the checksum is correct, the connection record is updated and an acknowledgement is sent back. The general scheme of first making a quick check to see if the header is what is expected, and having a special procedure to handle that case, is called **header prediction**. Many TCP implementations use it. When this optimization and all the other ones discussed in this chapter are used together, it is possible to get TCP to run at 90 percent of the speed of a local memory-to-memory copy, assuming the network itself is fast enough.

Ex. 1006.585 (Tanenbaum96); *see also* Paper 35 (Reply) at 2.

Thia was not theoretical and offered a practical design for a hardware bypass

are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

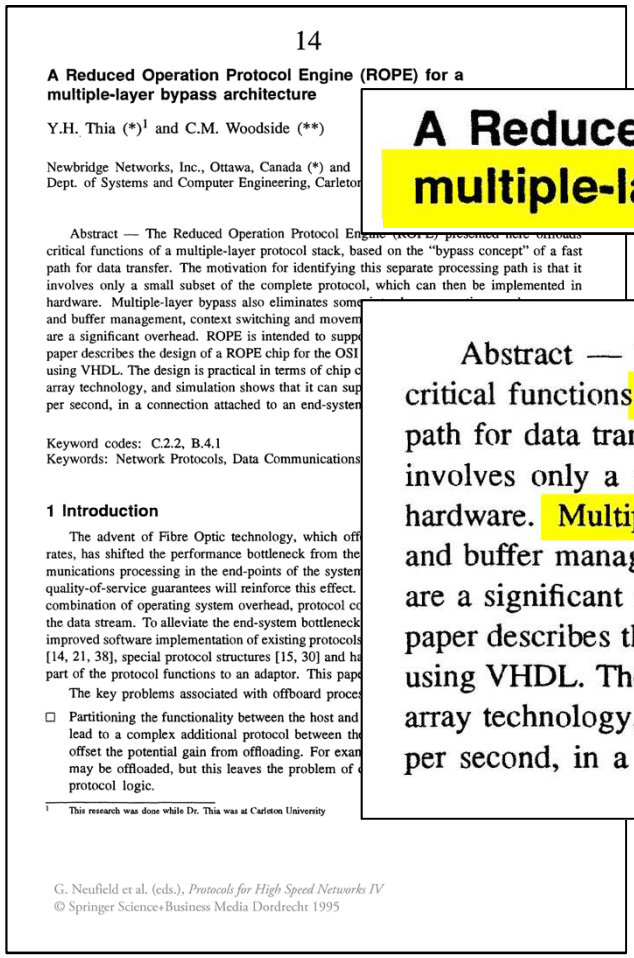
4.3 First Design: Design Steps

Figure 3 shows the steps followed in this study. There were three stages, a behavioural model, a structural or RTL model, and a gate level design. These gave us two kinds of feasibility check, that the logic we specified will execute the protocol within the environment we envisage, and that the design is technically feasible, for instance in a reasonable chip area.

It can be concluded from this study that it is feasible to implement the bypass stack (at least for the transport and session layers) in VLSI and that the performance would be at least an order of magnitude higher than software protocol processing. The bypass system offloads the critical protocol functions and the associated non-protocol-specific functions onto a "Reduced Operation Protocol Engine" (ROPE). The gate count for the bypass chip can

Ex. 1015.001, .008, .013 (Thia);
see also Paper 35 (Reply) at 7-8; Ex.1399.033-.034 (Horst Reply Decl.).

Thia's teachings are not limited to OSI



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications

1 Introduction

The advent of Fibre Optic technology, which off
rates, has shifted the performance bottleneck from the
munications processing in the end-points of the system
quality-of-service guarantees will reinforce this effect.
combination of operating system overhead, protocol co
the data stream. To alleviate the end-system bottleneck
improved software implementation of existing protocols
[14, 21, 38], special protocol structures [15, 30] and ha
part of the protocol functions to an adaptor. This pape

The key problems associated with offboard proces

- Partitioning the functionality between the host and
lead to a complex additional protocol between the
offset the potential gain from offloading. For exam
may be offloaded, but this leaves the problem of
protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

Ex. 1015.001 (Thia); *see also* Paper 35 (Reply) at 5;
Ex. 1399.030-.031 (Horst Reply Decl.).

Thia's standard protocol stack (SPS) is a "multi-layer" stack, not an "OSI" stack

14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass arch

Y.H. Thia (*)[1] and C.M. Woo

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

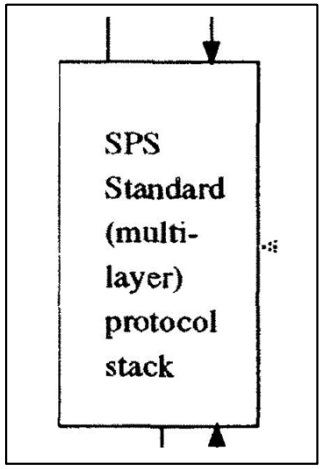
The key problems associated with offboard processing include:

- Partitioning the functionality bet
  lead to a complex additional pr
  offset the potential gain from off
  may be offloaded, but this leave
  protocol logic.

[1] This research was done while Dr. Thia was at C

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

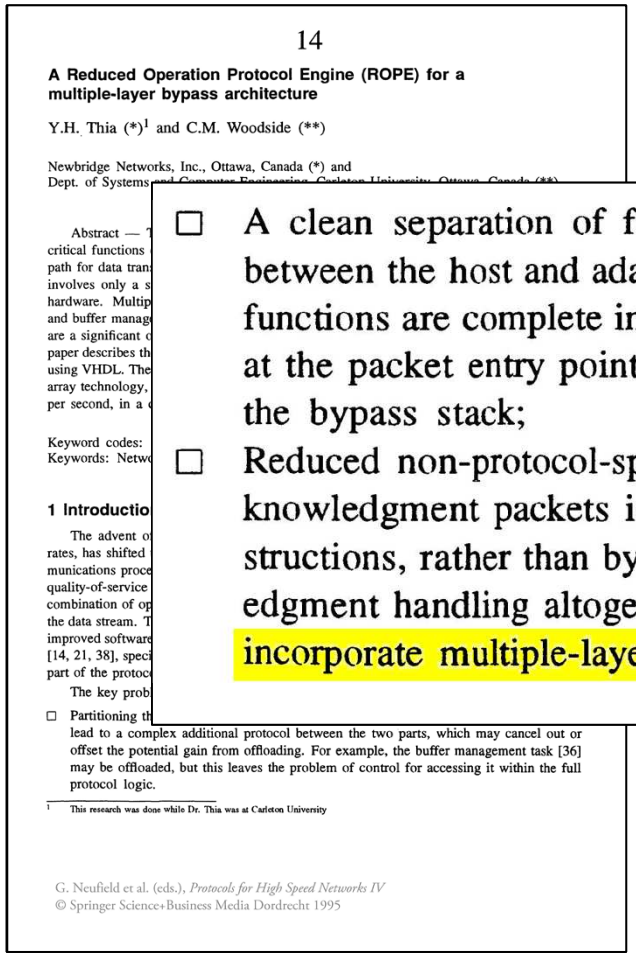
Figure 1 illustrates the architecture of a bypass implementation for any standard protocol.



without the bypass. The SPS may refer to a single layer or to multiple adjoining layers of a layered protocol stack. The bypass has 4 key components:

Ex. 1015.003 (Thia); *see also* Paper 35 (Reply) at 5;
Ex. 1399.030-.031 (Horst Reply Decl.).

Thia teaches that its bypass offload is more than one multi-layer stack



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems

Abstract —
critical functions
path for data tran
involves only a s
hardware. Multip
and buffer manag
are a significant
paper describes th
using VHDL. The
array technology,
per second, in a

Keyword codes:
Keywords: Netw

1 Introductio

The advent o
rates, has shifted
munications proce
quality-of-service
combination of op
the data stream. T
improved software
[14, 21, 38], speci
part of the protoc

The key prob

- Partitioning th

lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

- A clean separation of functionality requiring only a simple protocol to communicate between the host and adaptor is desired, and is provided by a bypass. Its particular set of functions are complete in themselves and have a focussed interface with the host software at the packet entry point. There is relatively infrequent switching between the SPS and the bypass stack;
- Reduced non-protocol-specific processing overhead. For example the processing of acknowledgment packets is dominated by interrupt handling, typically a few hundred instructions, rather than by the protocol processing itself. Our approach removes acknowledgment handling altogether from the host. Also, the bypass system can be extended to incorporate multiple-layer stacks and remove overhead that way;

Ex. 1015.005 (Thia); *see also* Paper 35 (Reply) at 6-7;
Ex.1399.030-.031 (Horst Reply Decl.).

TCP/IP and OSI were widely understood to be very similar

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION
Petitioner

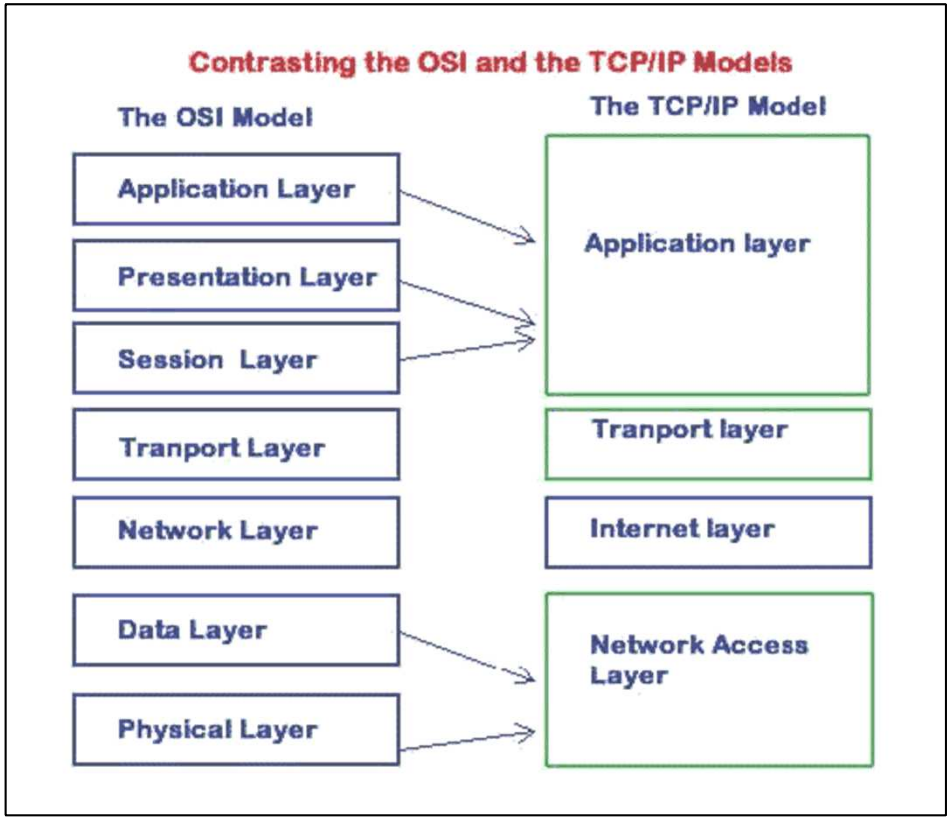
v.

ALACRITECH, INC.
Patent Owner

Case IPR. No. **Unassigned**
U.S. Patent No. 8,850,948
Title: INTELLIGENT NETWORK INTERFACE SYSTEM AND METHOD FOR PROTOCOL PROCESSING

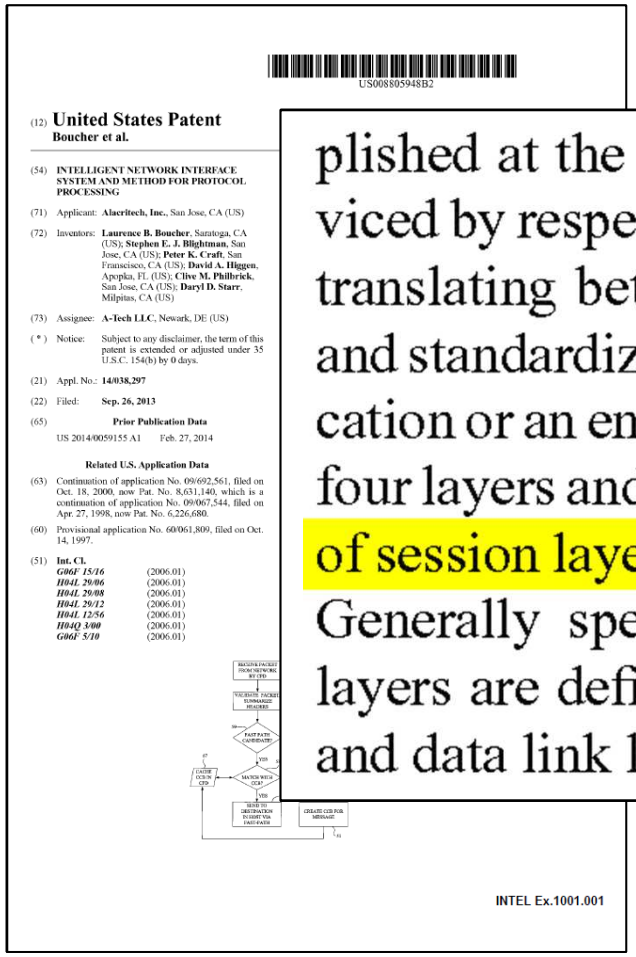
Declaration of Robert Horst, Ph.D. in Support of Petition for *Inter Partes* Review of U.S. Patent No. 8,850,948

INTEL Ex.1003.001



Ex. 1003.013 (Horst Decl.);
see also Paper 35 (Reply) at 6.

The 948 Patent admits that TCP/IP layers correspond to OSI layers



US008805948B2

(12) **United States Patent**
Boucher et al.

(54) **INTELLIGENT NETWORK INTERFACE SYSTEM AND METHOD FOR PROTOCOL PROCESSING**

(71) Applicant: **Alacritech, Inc.**, San Jose, CA (US)

(72) Inventors: **Laurence B. Boucher**, Saratoga, CA (US); **Stephen E. J. Blightman**, San Jose, CA (US); **Peter K. Craft**, San Francisco, CA (US); **David A. Higgen**, Apopka, FL (US); **Clive M. Philbrick**, San Jose, CA (US); **Daryl D. Starr**, Milpitas, CA (US)

(73) Assignee: **A-Tech LLC**, Newark, DE (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/038,297**

(22) Filed: **Sep. 26, 2013**

(65) **Prior Publication Data**

US 2014/0059155 A1 Feb. 27, 2014

Related U.S. Application Data

(63) Continuation of application No. 09/692,561, filed on Oct. 18, 2000, now Pat. No. 8,631,140, which is a continuation of application No. 09/067,544, filed on Apr. 27, 1998, now Pat. No. 6,226,680.

(60) Provisional application No. 60/061,809, filed on Oct. 14, 1997.

(51) **Int. Cl.**

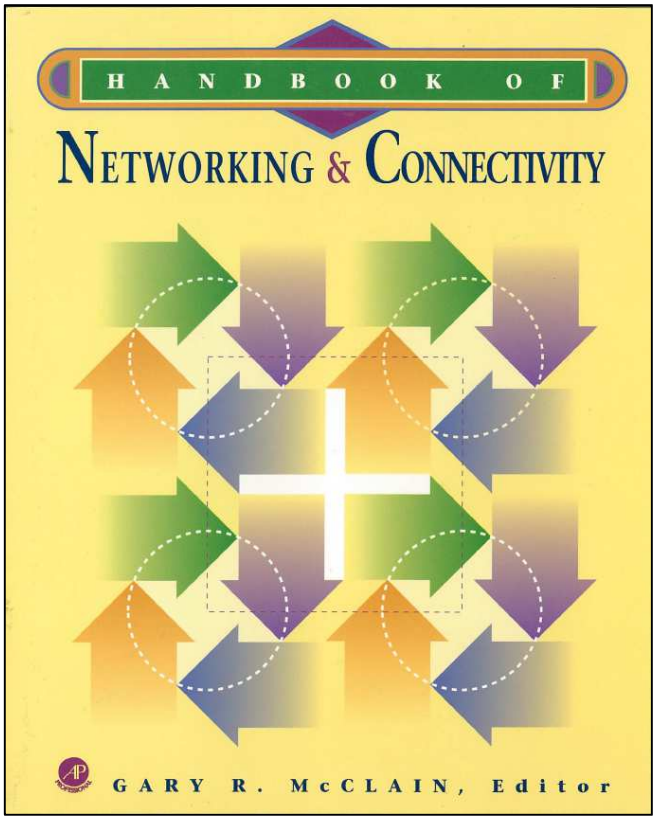
G06F 15/16	(2006.01)
H04L 29/06	(2006.01)
H04L 29/08	(2006.01)
H04L 29/12	(2006.01)
H04L 12/56	(2006.01)
H04Q 3/00	(2006.01)
G06F 5/10	(2006.01)

INTEL Ex.1001.001

plished at the presentation level. Application layers are serviced by respective presentation layers, the application layers translating between programs particular to individual hosts and standardized programs for presentation to either an application or an end user. The TCP/IP standard includes the lower four layers and application layers, but integrates the functions of session layers and presentation layers into adjacent layers. Generally speaking, application, presentation and session layers are defined as upper layers, while transport, network and data link layers are defined as lower layers.

Ex. 1001 at 2:10-19 (948 Patent);
see also Paper 35 (Reply) at 6; Ex.1399.031-.032 (Horst Reply Decl.).

Thia's disclosure of FDDI does not preclude TCP/IP as it was commonly known to use them together



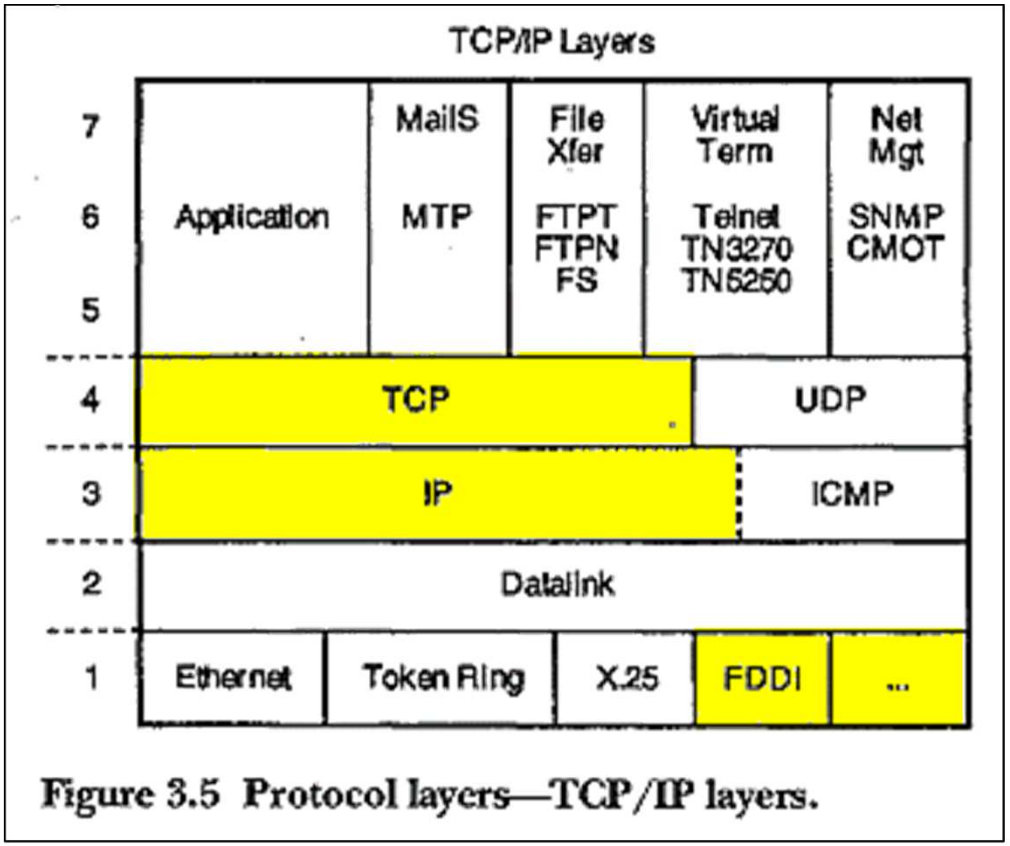


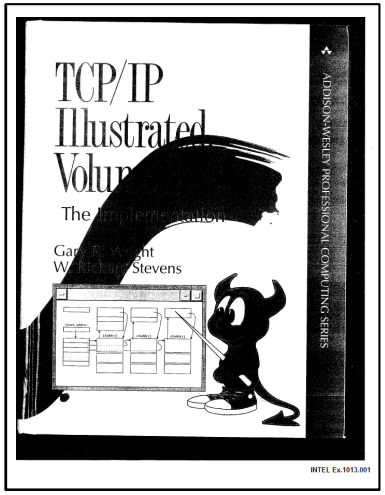
Figure 3.5 Protocol layers—TCP/IP layers.

Ex. 1252.022-.023 (McClain);
see also Paper 35 (Reply) at 6.

948 Patent: Disputes

1. A POSA would have been motivated to combine Thia and Tanenbaum96 (and Stevens2)
   a. Tanenbaum96 does not teach away from the combination
   b. The trend towards TCP/IP in the 1990s would motivate combining Thia's bypass architecture with TCP/IP
   c. A POSA would have understood that Thia's teachings are applicable to TCP/IP
   d. It would have been obvious to combine Stevens2 with Thia and Tanenbaum96

Each discloses a bypass/fast-path based on TCP/IP header prediction



28.4 Header Prediction

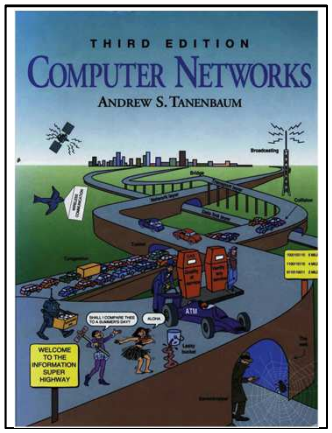
We now continue with the code in `tcp_input`, from where we left off in Figure 28.8. *Header prediction* was put into the 4.3BSD Reno release by Van Jacobson. The only description of the algorithm, other than the source code we're about to examine, is in [Jacobson 1990b], which is a copy of three slides showing the code.

Header prediction helps unidirectional data transfer by handling the two common cases.

1. If TCP is sending data, the next expected segment for this connection is an ACK for outstanding data.
2. If TCP is receiving data, the next expected segment for this connection is the next in-sequence data segment.

In both cases a small set of tests determines if the next expected segment has been received, and if so, it is handled in-line, faster than the general processing that follows later in this chapter and the next.

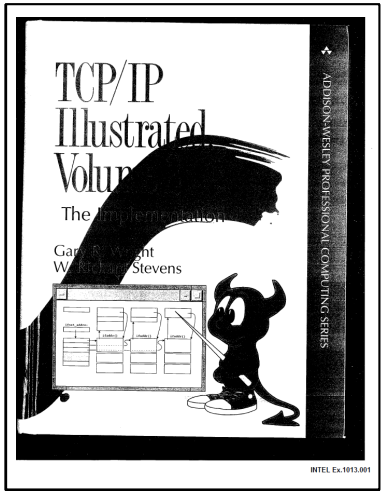
Ex. 1013.960-.962 (Stevens2); *see also* Paper 35 (Reply) at 7; Paper 2 (Petition) at 56-60; Ex.1399.033-.034 (Horst Reply Decl.); Ex.1003.078-.080 (Horst Decl.).



The fast path updates the connection record and copies the data to the user. While it is copying, it also computes the checksum, eliminating an extra pass over the data. If the checksum is correct, the connection record is updated and an acknowledgement is sent back. The general scheme of first making a quick check to see if the header is what is expected, and having a special procedure to handle that case, is called **header prediction**. Many TCP implementations use it. When this optimization and all the other ones discussed in this chapter are used together, it is possible to get TCP to run at 90 percent of the speed of a local memory-to-memory copy, assuming the network itself is fast enough.

Ex. 1006.585 (Tanenbaum96); *see also* Paper 35 (Reply) at 2.

Each discloses a bypass/fast-path based on TCP/IP header prediction



28.4 Header Prediction

We now continue with the code in `tcp_input`, from where we left off in Figure 28.8. *Header prediction* was put into the 4.3BSD Reno release by Van Jacobson. The only description of the algorithm, other than the source code we're about to examine, is in [Jacobson 1990b], which is a copy of three slides showing the code.

Header prediction helps unidirectional data transfer by handling the two common cases.

1. If TCP is sending data, the next expected segment for this connection is an ACK for outstanding data.
2. If TCP is receiving data, the next expected segment for this connection is the next in-sequence data segment.

In both cases a small set of tests determines if the next expected segment has been received, and if so, it is handled in-line, faster than the general processing that follows later in this chapter and the next.

Ex. 1013.960-.962 (Stevens2); *see also* Paper 35 (Reply) at 7; Paper 2 (Petition) at 56-60; Ex.1399.033-.034 (Horst Reply Decl.); Ex.1003.078-.080 (Horst Decl.).

14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University.

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

This paper presents a feasibility study for a new approach to hardware assistance. It combines the relatively simple operations needed for data transfer across multiple layers and provides a hardware "fast path" for them, which will be efficient for bulk data transfer. It is based on the "protocol bypass concept" [37] which is a generalization of Jacobson's "Header Prediction" algorithm [20] for TCP/IP. Bypass solves the problems identified above, which may limit the use of offboard processing, by implementing an entire service through all layers for certain cases. This simplifies the interface between the host and the adaptor chip and minimizes their interaction, which is supported by an access test, some DMA processing and a simple command protocol. The chip design based on bypassing is called ROPE, for

Ex. 1015.002 (Thia); *see also* Paper 35 at 2.

948 Patent: Disputes

1. A POSA would have been motivated to combine Thia and Tanenbaum96 (and Stevens2)

2. **The prior art combinations disclose the limitations of the challenged claims of the 948 Patent**

948 Patent: Disputes

2. The prior art combinations disclose the limitations of the challenged claims of the 948 Patent

   a. The combination discloses a network interface checking whether packets are IP fragmented

   b. The combination discloses checking whether "packets" have certain exception conditions / the combination discloses the protocol stack processing exception conditions

   c. The combination discloses bypassing host protocol stack processing and storing data from packets without exception conditions (Board previously found that Thia and Tanenbaum96 teach this)

948 Patent: Claims 1, 17

The invention claimed is:

1. A method for network communication by a host computer having a network interface that is connected to the host by an input/output bus, the method comprising:

running, on the host computer, a protocol processing stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, with an application layer running above the TCP layer;

initializing, by the host computer, a TCP connection that is defined by source and destination IP addresses and source and destination TCP ports;

receiving, by the network interface, first and second packets, wherein the first packet has a first TCP header and contains first payload data for the application, and the second packet has a second TCP header and contains second payload data for the application;

checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order;

if the first packet has any of the exception conditions, then protocol processing the first TCP header by the protocol processing stack;

if the second packet has any of the exception conditions, then protocol processing the second TCP header by the protocol processing stack;

if the packets do not have any of the exception conditions, then bypassing host protocol processing of the TCP headers and storing the first payload data and the second payload data together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the first payload data and the second payload data.

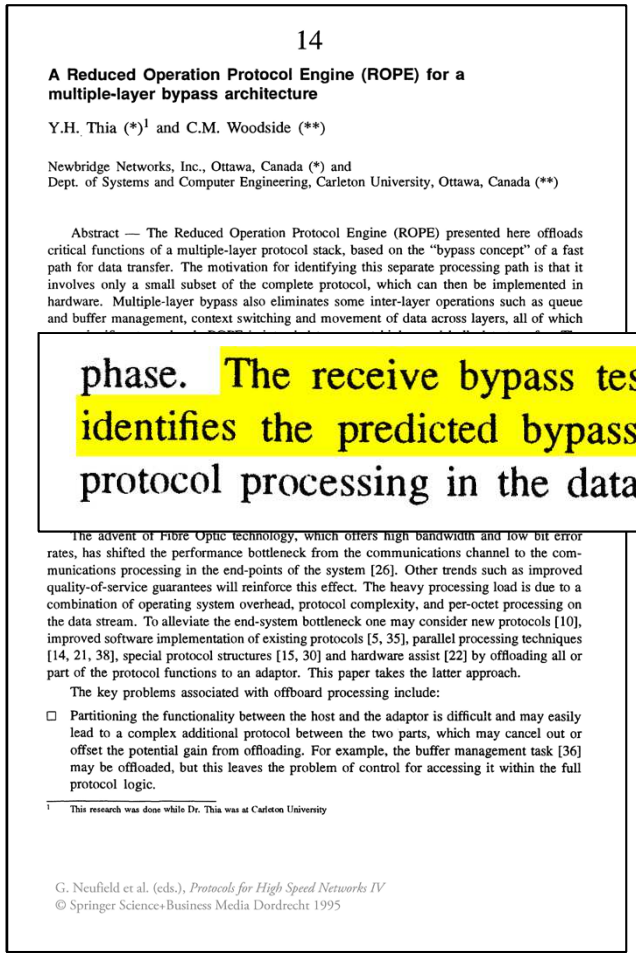
17. An apparatus for network communication, the apparatus comprising:

a host computer running a protocol stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, the protocol stack adapted to establish a TCP connection for an application layer running above the TCP layer, the TCP connection being defined by source and destination IP addresses and source and destination TCP ports;

a network interface that is connected to the host computer by an input/output bus, the network interface adapted to parse the headers of received packets to determine whether the headers have the IP addresses and TCP ports that define the TCP connection and to check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order, the network interface having logic that directs any of the received packets that have the exception conditions to the protocol stack for processing, and directs the received packets that do not have any of the exception conditions to have their headers removed and their payload data stored together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the payload data that came from different packets of the received packets.

Ex. 1001 (948 Patent) at Claim 1, Claim 7.

Thia + Tanenbaum96 teaches checking for fragmentation in fast-path test



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which

phase. The receive bypass test matches the incoming PDU headers with a template that identifies the predicted bypassable headers. The bypass stack performs all the relevant protocol processing in the data transfer phase. The shared data are used to maintain state

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

Thia's RX bypass test checks PDU headers to determine if packets are bypassable

Ex. 1015.003 (Thia); *see also* Paper 35 (Reply) at 8; Paper 2 at 75 (Petition).

Thia + Tanenbaum96 teaches checking for fragmentation in fast-path test

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

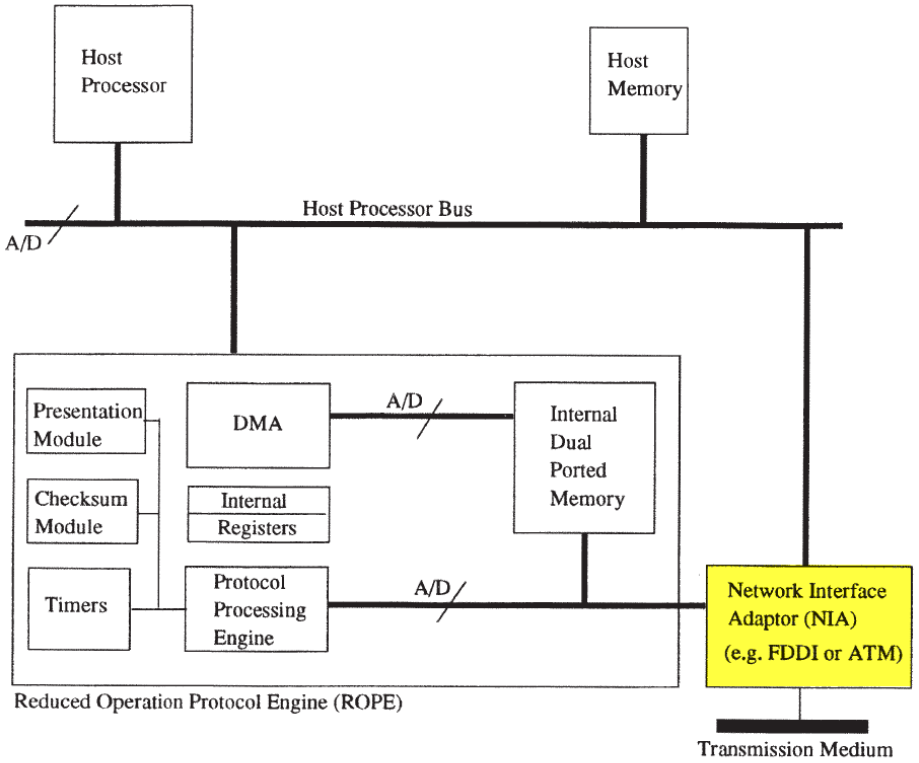
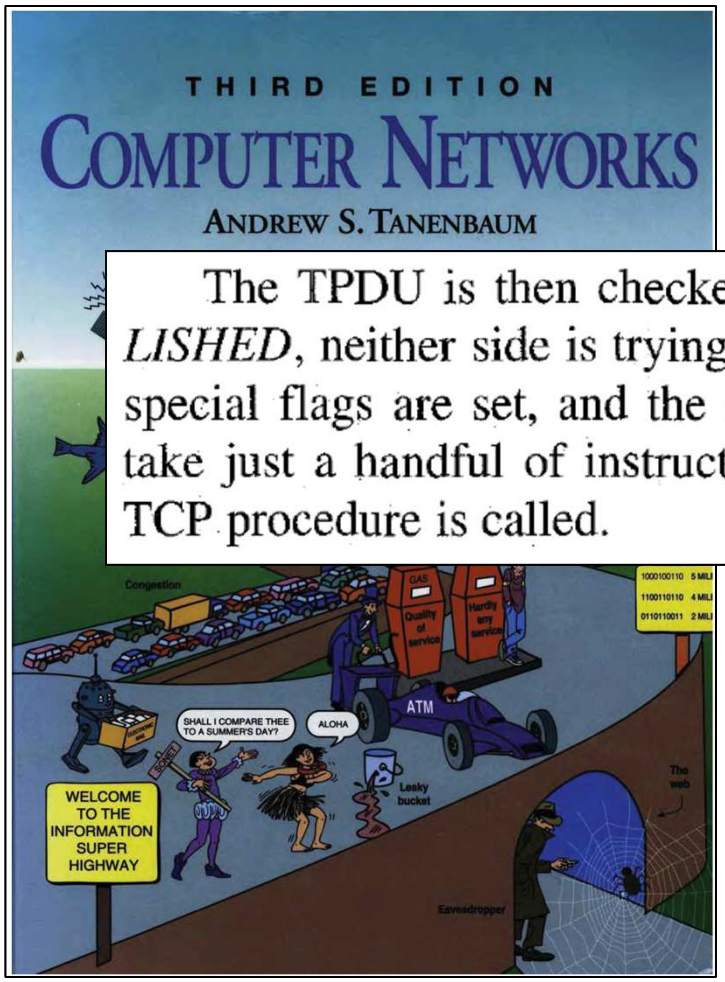


Figure 2 Block Diagram of VLSI bypass system

Ex. 1015.007 (Thia); *see also* Paper 35 (Reply) at 8; Paper 2 at 75 (Petition).

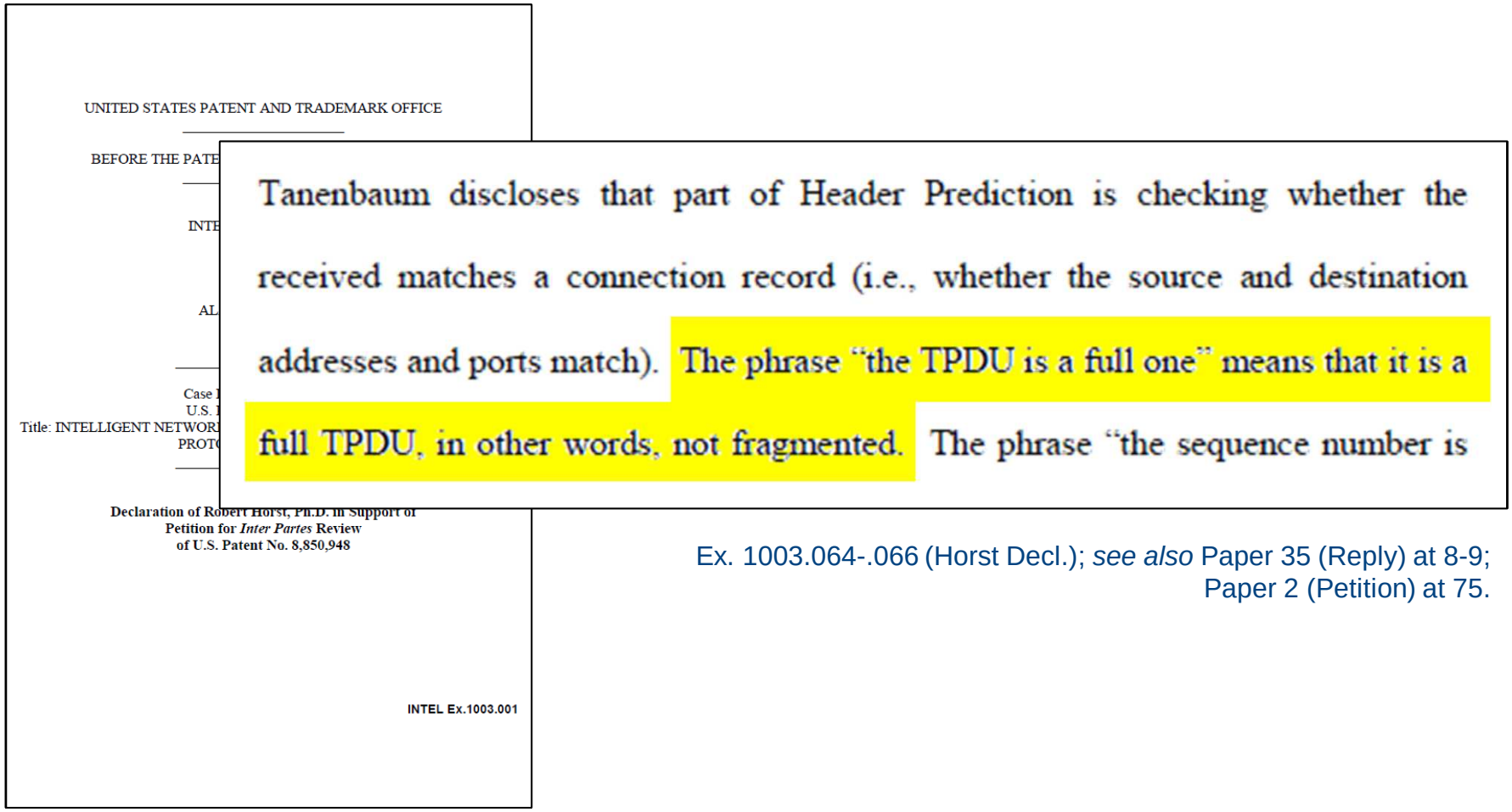
Thia + Tanenbaum96 teaches checking for fragmentation in fast-path test



The TPDU is then checked to see if it is a normal one: the state is *ESTABLISHED*, neither side is trying to close the connection, the TPDU is a full one, no special flags are set, and the sequence number is the one expected. These tests take just a handful of instructions. If all conditions are met, a special fast path TCP procedure is called.

Ex. 1006.585 (Tanenbaum96);
see also Paper 35 (Reply) at 8.

Undisputed: A POSA would understand "the TPDU is a full one" to mean it is not fragmented



UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATE

INTE

AL

Case I
U.S. I
Title: INTELLIGENT NETWOR
PROT

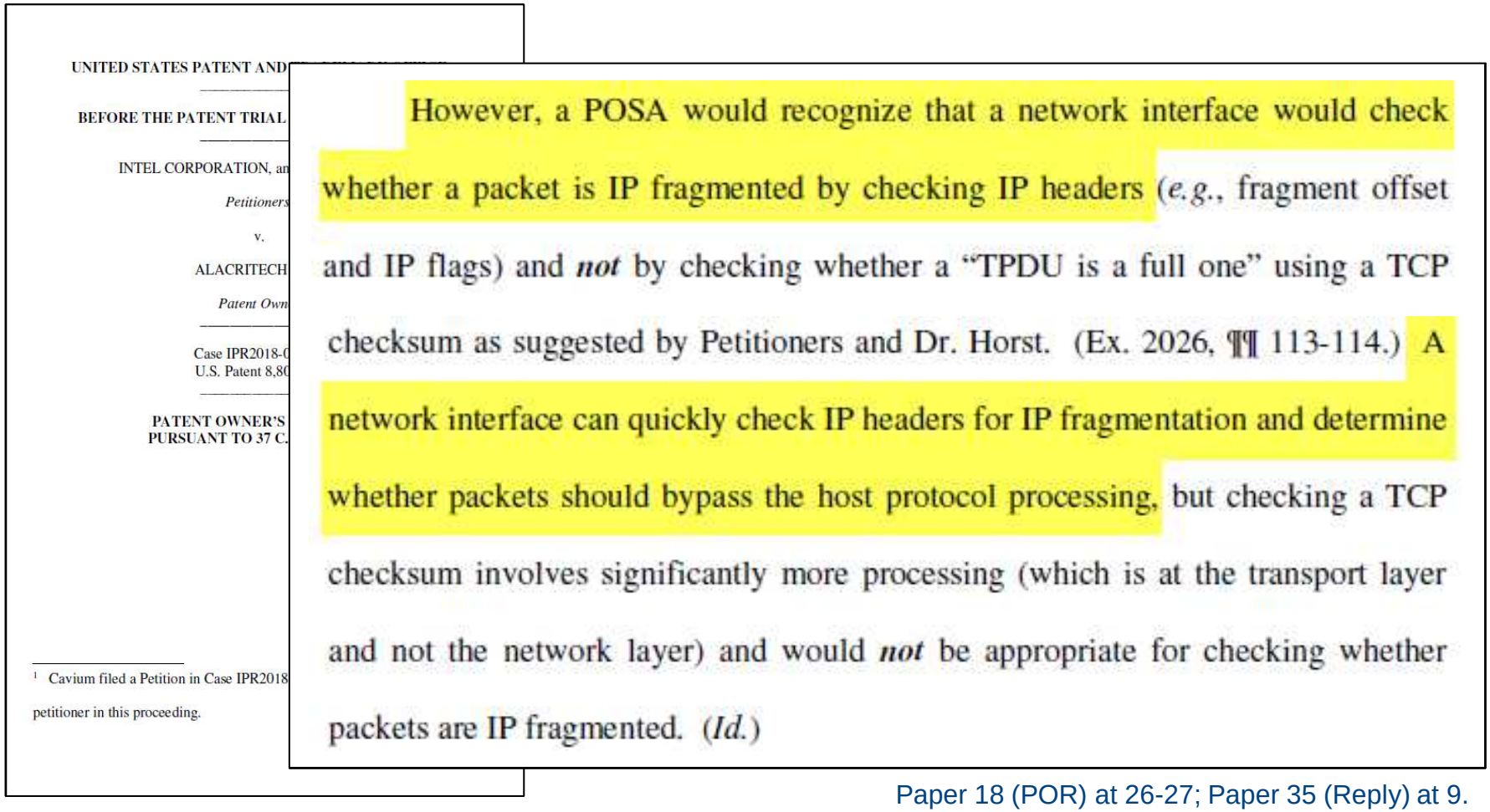
Declaration of Robert Horst, Ph.D. in Support of Petition for *Inter Partes* Review of U.S. Patent No. 8,850,948

INTEL Ex.1003.001

Tanenbaum discloses that part of Header Prediction is checking whether the received matches a connection record (i.e., whether the source and destination addresses and ports match). The phrase "the TPDU is a full one" means that it is a full TPDU, in other words, not fragmented. The phrase "the sequence number is

Ex. 1003.064-.066 (Horst Decl.); *see also* Paper 35 (Reply) at 8-9; Paper 2 (Petition) at 75.

PO admits a POSA would know how to check for fragmentation



UNITED STATES PATENT AND

BEFORE THE PATENT TRIAL

INTEL CORPORATION, an

Petitioners

v.

ALACRITECH

Patent Own

Case IPR2018-0
U.S. Patent 8,80

PATENT OWNER'S
PURSUANT TO 37 C.

[1] Cavium filed a Petition in Case IPR2018

petitioner in this proceeding.

However, a POSA would recognize that a network interface would check whether a packet is IP fragmented by checking IP headers (*e.g.*, fragment offset and IP flags) and ***not*** by checking whether a "TPDU is a full one" using a TCP checksum as suggested by Petitioners and Dr. Horst. (Ex. 2026, ¶¶ 113-114.) A network interface can quickly check IP headers for IP fragmentation and determine whether packets should bypass the host protocol processing, but checking a TCP checksum involves significantly more processing (which is at the transport layer and not the network layer) and would ***not*** be appropriate for checking whether packets are IP fragmented. (*Id.*)

Paper 18 (POR) at 26-27; Paper 35 (Reply) at 9.

948 Patent: Disputes

2. The prior art combinations disclose the limitations of the challenged claims of the 948 Patent

   a. The combination discloses a network interface checking whether packets are IP fragmented

   b. The combination discloses checking whether "packets" have certain exception conditions / the combination discloses the protocol stack processing exception conditions

   c. The combination discloses bypassing host protocol stack processing and storing data from packets without exception conditions (Board previously found that Thia and Tanenbaum96 teach this)

948 Patent: Claims 1, 17

The invention claimed is:

1. A method for network communication by a host computer having a network interface that is connected to the host by an input/output bus, the method comprising:

running, on the host computer, a protocol processing stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, with an application layer running above the TCP layer;

initializing, by the host computer, a TCP connection that is defined by source and destination IP addresses and source and destination TCP ports;

receiving, by the network interface, first and second packets, wherein the first packet has a first TCP header and contains first payload data for the application, and the second packet has a second TCP header and contains second payload data for the application;

checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order;

if the first packet has any of the exception conditions, then protocol processing the first TCP header by the protocol processing stack;

if the second packet has any of the exception conditions, then protocol processing the second TCP header by the protocol processing stack;

if the packets do not have any of the exception conditions, then bypassing host protocol processing of the TCP headers and storing the first payload data and the second payload data together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the first payload data and the second payload data.

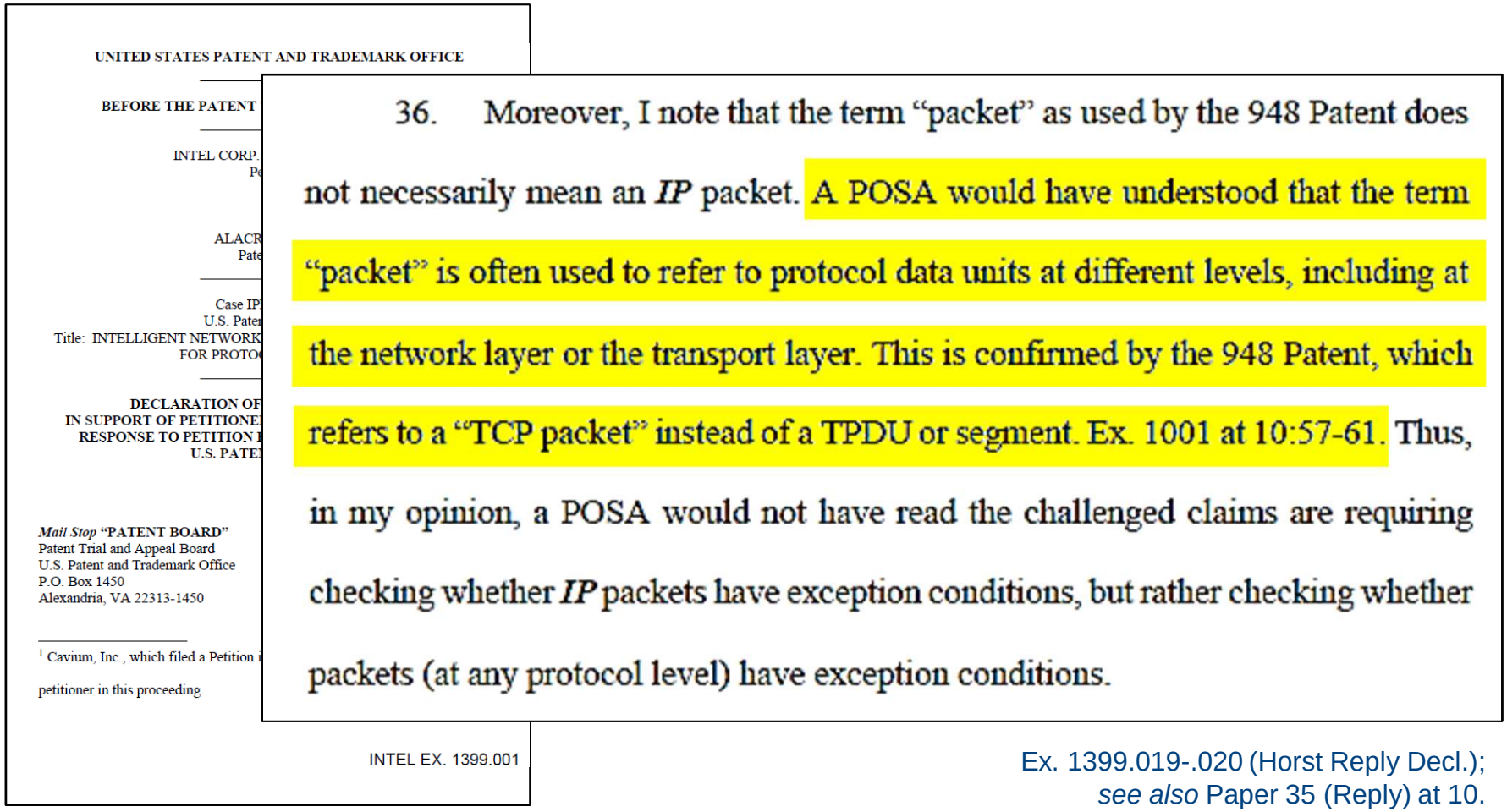
17. An apparatus for network communication, the apparatus comprising:

a host computer running a protocol stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, the protocol stack adapted to establish a TCP connection for an application layer running above the TCP layer, the TCP connection being defined by source and destination IP addresses and source and destination TCP ports;

a network interface that is connected to the host computer by an input/output bus, the network interface adapted to parse the headers of received packets to determine whether the headers have the IP addresses and TCP ports that define the TCP connection and to check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order, the network interface having logic that directs any of the received packets that have the exception conditions to the protocol stack for processing, and directs the received packets that do not have any of the exception conditions to have their headers removed and their payload data stored together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the payload data that came from different packets of the received packets.

Ex. 1001 (948 Patent) at Claim 1, Claim 7.

A POSA would not have understood "packet" to be limited to an IP packet



UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT

INTEL CORP.
Pe

ALACR
Pate

Case IP
U.S. Pater
Title: INTELLIGENT NETWORK
FOR PROTO

DECLARATION OF
IN SUPPORT OF PETITIONE
RESPONSE TO PETITION
U.S. PATE

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

[1] Cavium, Inc., which filed a Petition i

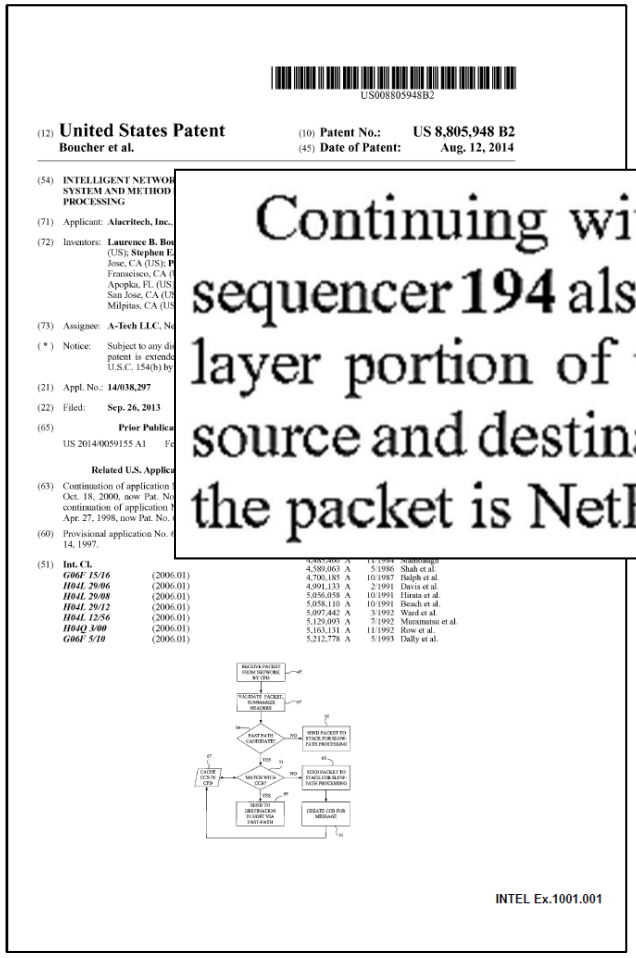
petitioner in this proceeding.

INTEL EX. 1399.001

36. Moreover, I note that the term "packet" as used by the 948 Patent does not necessarily mean an *IP* packet. A POSA would have understood that the term "packet" is often used to refer to protocol data units at different levels, including at the network layer or the transport layer. This is confirmed by the 948 Patent, which refers to a "TCP packet" instead of a TPDU or segment. Ex. 1001 at 10:57-61. Thus, in my opinion, a POSA would not have read the challenged claims are requiring checking whether *IP* packets have exception conditions, but rather checking whether packets (at any protocol level) have exception conditions.

Ex. 1399.019-.020 (Horst Reply Decl.);
see also Paper 35 (Reply) at 10.

948 Patent refers to a “TCP packet” instead of a TPDU or segment; so “packet” not limited to IP



US008805948B2

(12) **United States Patent**
Boucher et al.

(10) **Patent No.: US 8,805,948 B2**
(45) **Date of Patent: Aug. 12, 2014**

(54) **INTELLIGENT NETWOR… SYSTEM AND METHOD … PROCESSING**

(71) Applicant: **Alacritech, Inc.,**

(72) Inventors: **Laurence B. Bo…** (US); **Stephen E…** Jose, CA (US); **P…** Francisco, CA (… Apopka, FL (US… San Jose, CA (US… Milpitas, CA (US…

(73) Assignee: **A-Tech LLC**, No…

(*) Notice: Subject to any dis… patent is extende… U.S.C. 154(b) by …

(21) Appl. No.: **14/038,297**

(22) Filed: **Sep. 26, 2013**

(65) **Prior Publica…**

US 2014/0059155 A1 Fe…

Related U.S. Applica…

(63) Continuation of application … Oct. 18, 2000, now Pat. No… continuation of application … Apr. 27, 1998, now Pat. No. …

(60) Provisional application No. … 14, 1997.

(51) **Int. Cl.**

G06F 15/16	(2006.01)
H04L 29/06	(2006.01)
H04L 29/08	(2006.01)
H04L 29/12	(2006.01)
H04L 12/56	(2006.01)
H04Q 3/00	(2006.01)
G06F 5/10	(2006.01)

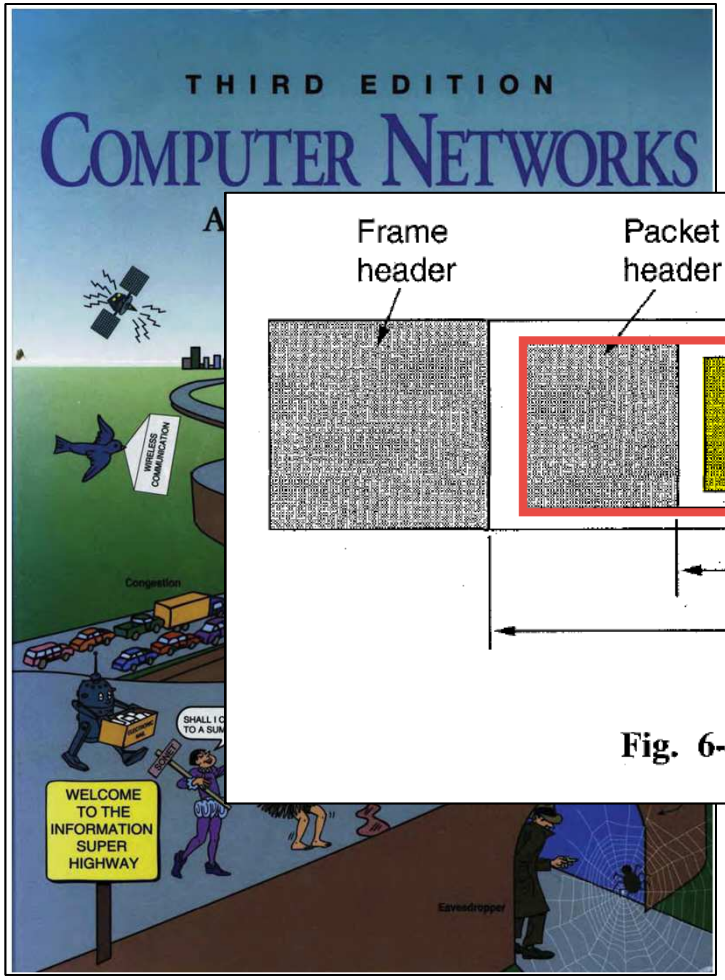
4,589,063	A	5/1986	Shah et al.
4,700,185	A	10/1987	Balph et al.
4,991,133	A	2/1991	Davis et al.
5,056,058	A	10/1991	Hirata et al.
5,058,110	A	10/1991	Beach et al.
5,097,442	A	3/1992	Ward et al.
5,129,093	A	7/1992	Muramatsu et al.
5,163,131	A	11/1992	Row et al.
5,212,778	A	5/1993	Dally et al.

INTEL Ex.1001.001

Continuing with the example of a TCP packet, transport sequencer **194** also analyzes the first few bytes in the transport layer portion of the header to determine, in part, the TCP source and destination ports for the message, such as whether the packet is NetBios or other protocols. Byte **12** of the TCP

Ex. 1001 (948 Patent.) at 10:57-61;
see also Paper 35 (Reply) at 10.

Even if "packet" meant "IP packet," PO ignores that a TPDU/segment is part of an IP packet



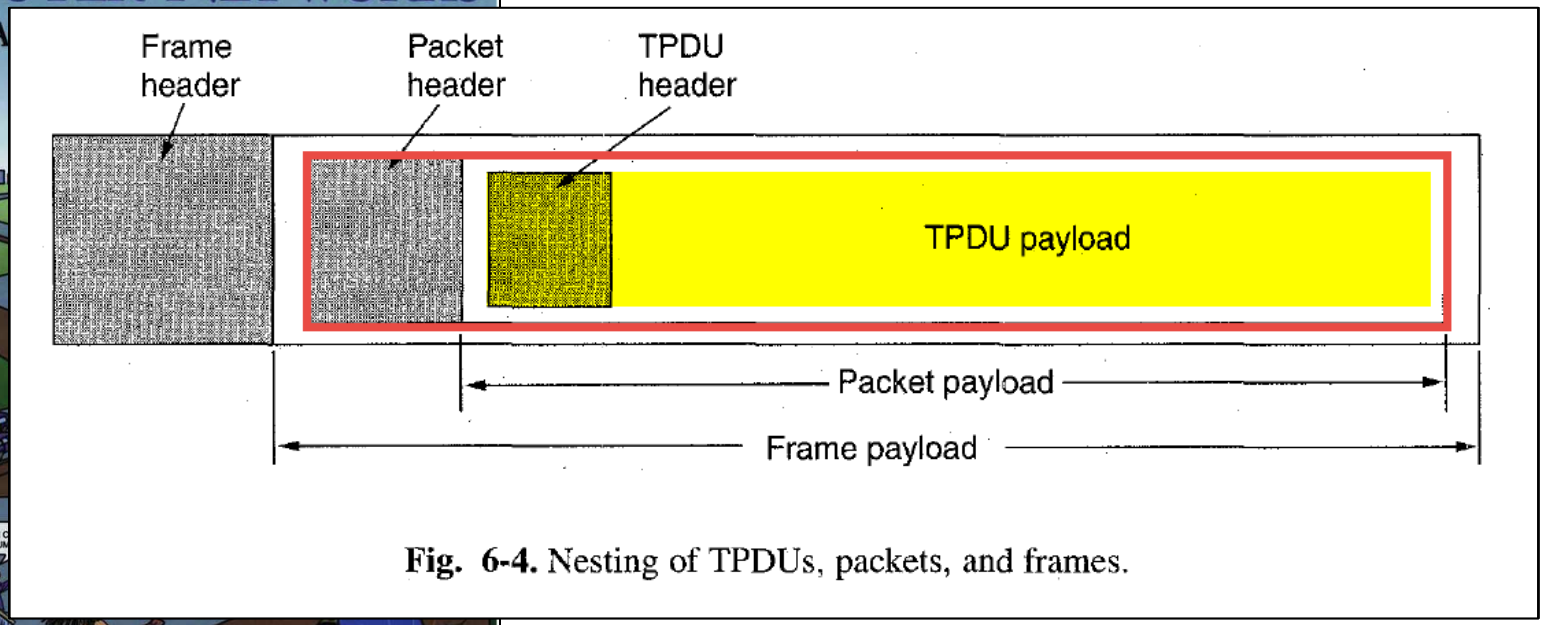


Fig. 6-4. Nesting of TPDUs, packets, and frames.

Ex. 1006.503 (Tanenbaum96);
see also Paper 35 (Reply) at 10,
Ex. 1003.015-.017 (Horst Decl.).

948 Patent: Claims 1, 17

The invention claimed is:

1. A method for network communication by a host computer having a network interface that is connected to the host by an input/output bus, the method comprising:

running, on the host computer, a protocol processing stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, with an application layer running above the TCP layer;

initializing, by the host computer, a TCP connection that is defined by source and destination IP addresses and source and destination TCP ports;

receiving, by the network interface, first and second packets, wherein the first packet has a first TCP header and contains first payload data for the application, and the second packet has a second TCP header and contains second payload data for the application;

checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order;

if the first packet has any of the exception conditions, then protocol processing the first TCP header by the protocol processing stack;

if the second packet has any of the exception conditions, then protocol processing the second TCP header by the protocol processing stack;

if the packets do not have any of the exception conditions, then bypassing host protocol processing of the TCP headers and storing the first payload data and the second payload data together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the first payload data and the second payload data.

17. An apparatus for network communication, the apparatus comprising:

a host computer running a protocol stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, the protocol stack adapted to establish a TCP connection for an application layer running above the TCP layer, the TCP connection being defined by source and destination IP addresses and source and destination TCP ports;

a network interface that is connected to the host computer by an input/output bus, the network interface adapted to parse the headers of received packets to determine whether the headers have the IP addresses and TCP ports that define the TCP connection and to check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order, the network interface having logic that directs any of the received packets that have the exception conditions to the protocol stack for processing, and directs the received packets that do not have any of the exception conditions to have their headers removed and their payload data stored together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the payload data that came from different packets of the received packets.

Ex. 1001 (948 Patent) at Claim 1, Claim 7.

Checking a characteristic of a TPDU is checking whether the IP packet has that characteristic

depicted below, the FIN flag is part of the TCP header.

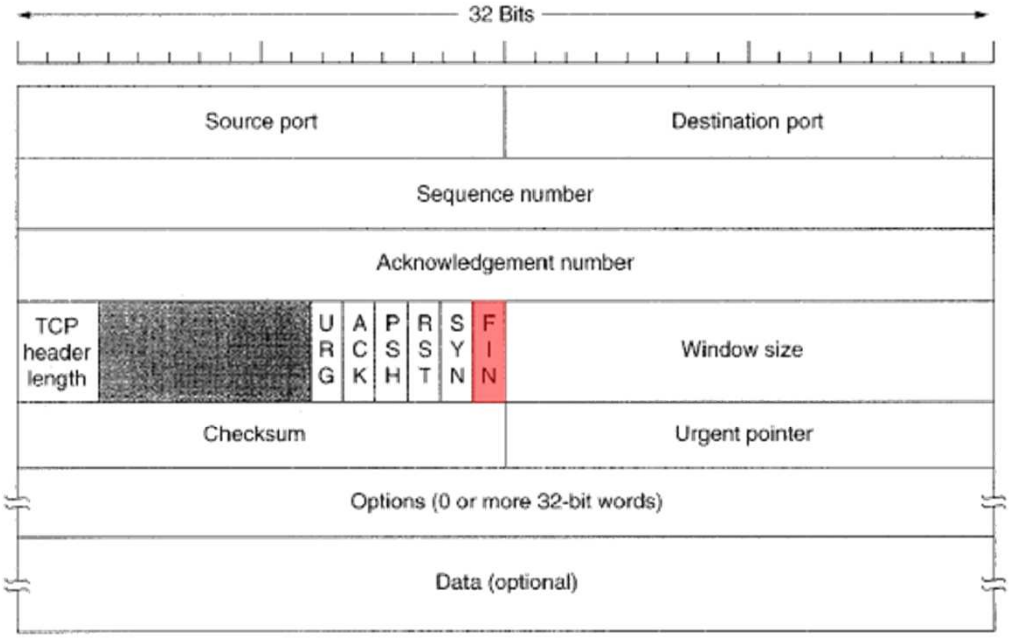


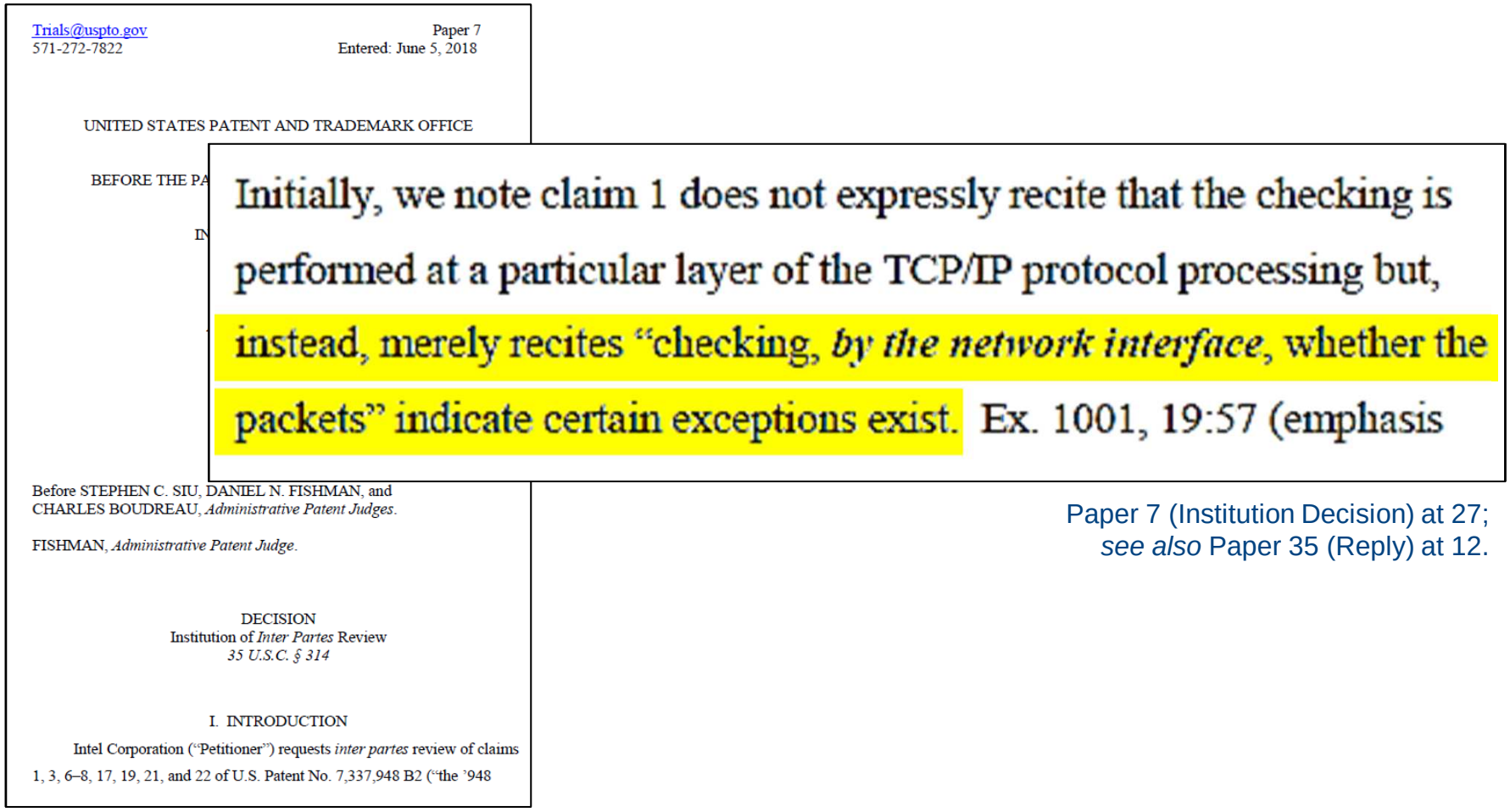
Fig. 6-24. The TCP header.

Ex. 1006, Tanenbaum96 at .544 (red shading added).

Thus, "checking … whether … *packets* have a FIN flag set," as required for example by claim 1, means checking the TCP header of ***TPDUs***, not the IP header.

Ex. 1399.017-.019 (Horst Reply Decl.); *see also* Paper 35 (Reply) at 11-12.

Institution Decision correctly noted the claims do not expressly recite checking at a particular layer



Trials@uspto.gov
571-272-7822

Paper 7
Entered: June 5, 2018

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PA

IN

Initially, we note claim 1 does not expressly recite that the checking is performed at a particular layer of the TCP/IP protocol processing but, instead, merely recites "checking, *by the network interface*, whether the packets" indicate certain exceptions exist. Ex. 1001, 19:57 (emphasis

Before STEPHEN C. SIU, DANIEL N. FISHMAN, and CHARLES BOUDREAU, *Administrative Patent Judges*.

FISHMAN, *Administrative Patent Judge*.

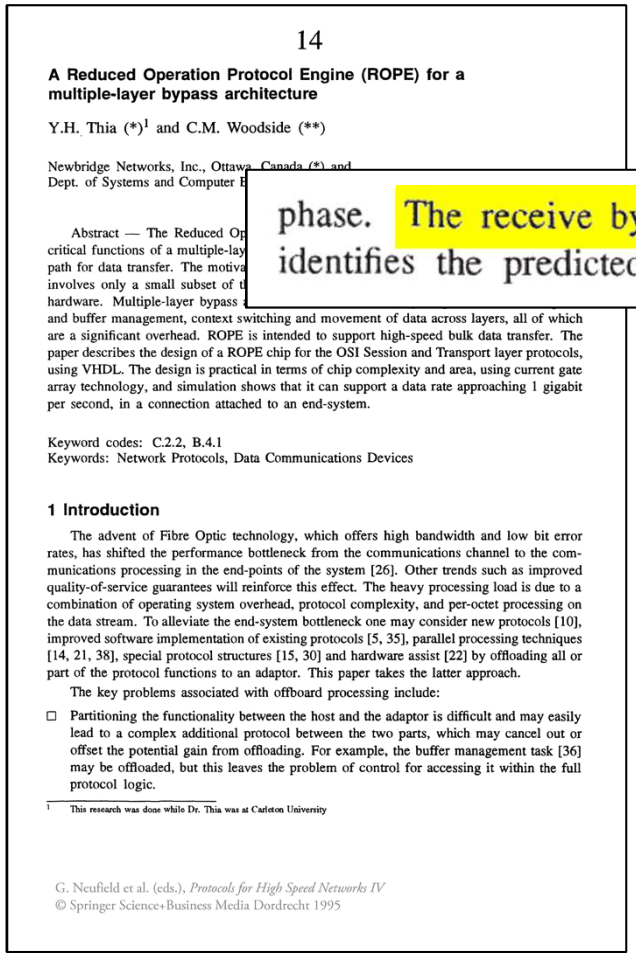
DECISION
Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Intel Corporation ("Petitioner") requests *inter partes* review of claims 1, 3, 6–8, 17, 19, 21, and 22 of U.S. Patent No. 7,337,948 B2 ("the '948

Paper 7 (Institution Decision) at 27;
see also Paper 35 (Reply) at 12.

Even if checking the network layer header were required (it is not), Thia does this



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer E

Abstract — The Reduced Op
critical functions of a multiple-lay
path for data transfer. The motiva
involves only a small subset of t
hardware. Multiple-layer bypass
and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

phase. The receive bypass test matches the incoming PDU headers with a template that identifies the predicted bypassable headers. The bypass stack performs all the relevant

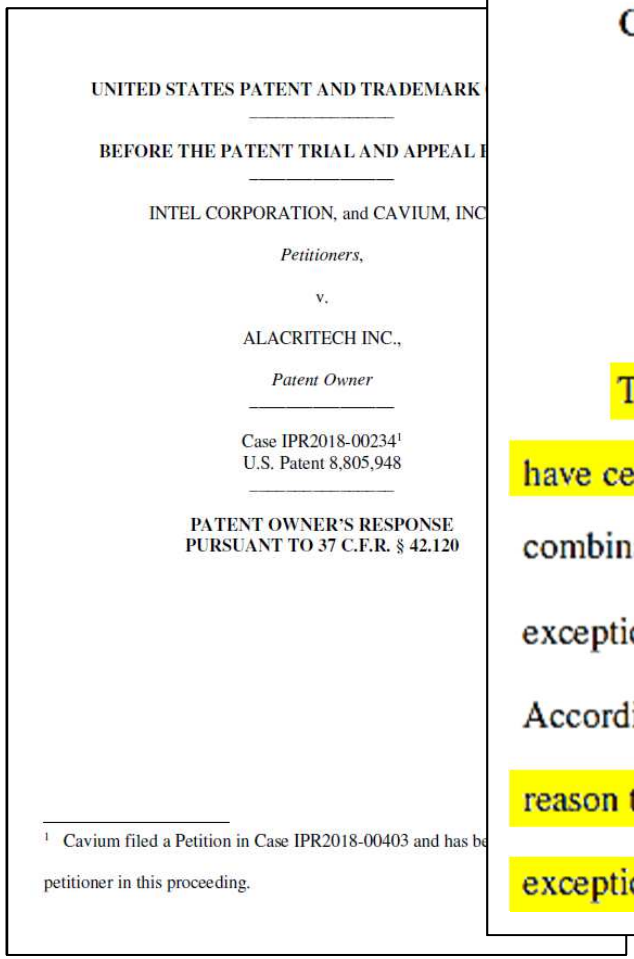
Ex. 1015.003 (Thia); *see also* Paper 35 (Reply) at 12 n.4; Paper 2 (Petition) at 77.

Even if checking the IP header were required (it is not), Header Prediction does this

Now let us look at fast path processing on the receiving side of Fig. 6-49. Step 1 is locating the connection record for the incoming TPDU. For ATM, finding the connection record is easy: the *VPI* field can be used as an index into the path table to find the virtual circuit table for that path and the *VCI* can be used as an index to find the connection record. For TCP, the connection record can be stored in a hash table for which some simple function of the two IP addresses and two ports is the key. Once the connection record has been located, both addresses and both ports must be compared to verify that the correct record has been found.

Ex.1006.584-.585 (Tanenbaum96); *see also* Paper 35 (Reply) at 12.

PO makes the same flawed arguments regarding the protocol processing limitations



UNITED STATES PATENT AND TRADEMARK

BEFORE THE PATENT TRIAL AND APPEAL

INTEL CORPORATION, and CAVIUM, INC

Petitioners,

v.

ALACRITECH INC.,

Patent Owner

Case IPR2018-00234[1]
U.S. Patent 8,805,948

PATENT OWNER'S RESPONSE
PURSUANT TO 37 C.F.R. § 42.120

[1] Cavium filed a Petition in Case IPR2018-00403 and has be

petitioner in this proceeding.

C. The Combination Does Not Show or Suggest "if the first packet has any of the exception conditions, then protocol processing the first TCP header by the protocol processing stack; [and] if the second packet has any of the exception conditions, then protocol processing the second TCP header by the protocol processing stack" (claim 1) / "the network interface having logic that directs any of the received packets that have the exception conditions to the protocol stack for processing" (claim 17)

These limitations are conditioned upon "check[ing] whether the packets have certain exception conditions." As explained above, in Sections VIII.A-B, the combination fails to disclose "check[ing] whether the packets have certain exception conditions, including . . . whether the packets are IP fragmented." Accordingly, the combination cannot disclose these limitations for at least the reason that they are conditioned upon "check[ing] whether the packets have certain exception conditions." (Ex. 2026, ¶ 121.)

Paper 18 (POR) at 38-50;
Paper 35 (Reply) at 12.

948 Patent: Disputes

2. The prior art combinations disclose the limitations of the challenged claims of the 948 Patent

   a. The combination discloses a network interface checking whether packets are IP fragmented

   b. The combination discloses checking whether “packets” have certain exception conditions / the combination discloses the protocol stack processing exception conditions

   c. The combination discloses bypassing host protocol stack processing and storing data from packets without exception conditions (Board previously found that Thia and Tanenbaum96 teach this)

948 Patent: Claims 1, 17

The invention claimed is:

1. A method for network communication by a host computer having a network interface that is connected to the host by an input/output bus, the method comprising:

running, on the host computer, a protocol processing stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, with an application layer running above the TCP layer;

initializing, by the host computer, a TCP connection that is defined by source and destination IP addresses and source and destination TCP ports;

receiving, by the network interface, first and second packets, wherein the first packet has a first TCP header and contains first payload data for the application, and the second packet has a second TCP header and contains second payload data for the application;

checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order;

if the first packet has any of the exception conditions, then protocol processing the first TCP header by the protocol processing stack;

if the second packet has any of the exception conditions, then protocol processing the second TCP header by the protocol processing stack;

if the packets do not have any of the exception conditions, then bypassing host protocol processing of the TCP headers and storing the first payload data and the second payload data together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the first payload data and the second payload data.

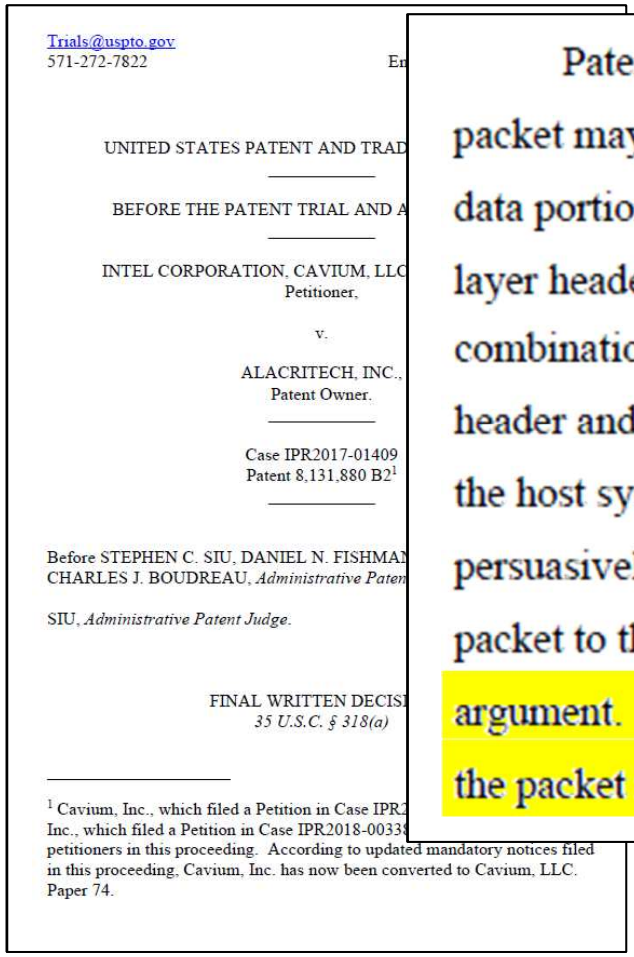
17. An apparatus for network communication, the apparatus comprising:

a host computer running a protocol stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, the protocol stack adapted to establish a TCP connection for an application layer running above the TCP layer, the TCP connection being defined by source and destination IP addresses and source and destination TCP ports;

a network interface that is connected to the host computer by an input/output bus, the network interface adapted to parse the headers of received packets to determine whether the headers have the IP addresses and TCP ports that define the TCP connection and to check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order, the network interface having logic that directs any of the received packets that have the exception conditions to the protocol stack for processing, and directs the received packets that do not have any of the exception conditions to have their headers removed and their payload data stored together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the payload data that came from different packets of the received packets.

Ex. 1001 (948 Patent) at Claim 1, Claim 7.

The Board previously found Thia and Tanenbaum96 teach storing data on the host without TCP headers



Trials@uspto.gov
571-272-7822

UNITED STATES PATENT AND TRAD

BEFORE THE PATENT TRIAL AND A

INTEL CORPORATION, CAVIUM, LLC
Petitioner,

v.

ALACRITECH, INC.,
Patent Owner.

Case IPR2017-01409
Patent 8,131,880 B2[1]

Before STEPHEN C. SIU, DANIEL N. FISHMA
CHARLES J. BOUDREAU, *Administrative Paten*

SIU, *Administrative Patent Judge*.

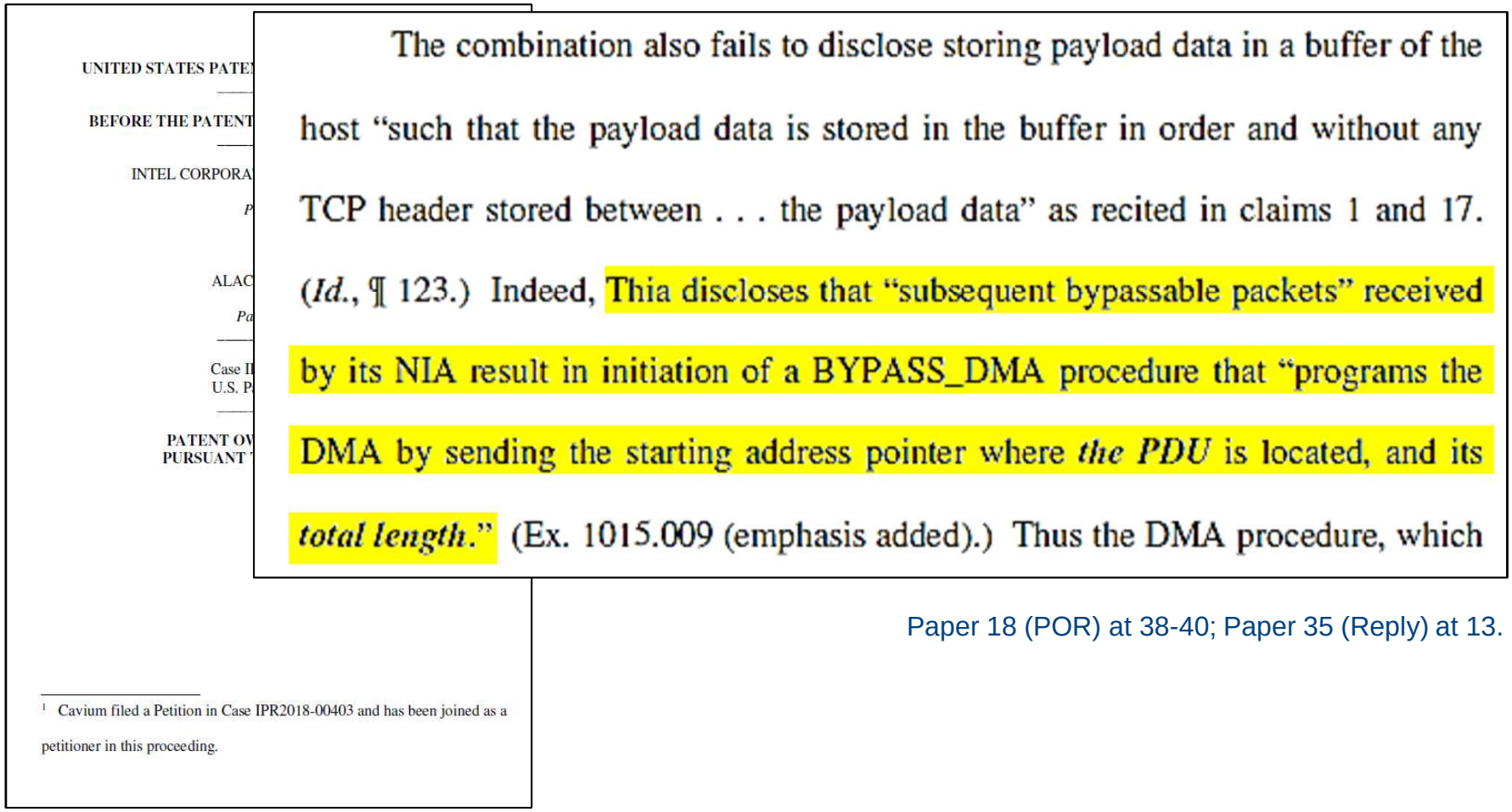
FINAL WRITTEN DECIS
35 U.S.C. § 318(a)

[1] Cavium, Inc., which filed a Petition in Case IPR
Inc., which filed a Petition in Case IPR2018-0033
petitioners in this proceeding. According to updated mandatory notices filed in this proceeding, Cavium, Inc. has now been converted to Cavium, LLC. Paper 74.

Patent Owner argues that Thia "merely states that the data portion of a packet may be copied" but "does not disclose or even suggest copying the data portion of a PDU ***without transferring*** the corresponding transport layer header." PO Resp. 46–47. However, as Petitioner points out, the combination of Thia and Tanenbaum discloses receiving a packet with a header and data portion and transferring the "data portion" of the packet to the host system memory. Patent Owner does not assert or demonstrate persuasively that Thia also discloses transferring the "header portion" of the packet to the host system memory. We are not persuaded by Patent Owner's argument. A skilled artisan would have understood that the data portion of the packet is transmitted to the host computer without the header.

IPR2017-01409 Paper 79 (FWD) at 10-11;
see also Paper 35 (Reply) at 13.

PO cites to Thia's TX (not RX) disclosures, to argue Thia transfers a whole PDU to the host



UNITED STATES PATE

BEFORE THE PATENT

INTEL CORPORA

P

ALAC

Pa

Case I
U.S. P

PATENT OV
PURSUANT

[1] Cavium filed a Petition in Case IPR2018-00403 and has been joined as a petitioner in this proceeding.

The combination also fails to disclose storing payload data in a buffer of the host "such that the payload data is stored in the buffer in order and without any TCP header stored between . . . the payload data" as recited in claims 1 and 17. (*Id.*, ¶ 123.) Indeed, Thia discloses that "subsequent bypassable packets" received by its NIA result in initiation of a BYPASS_DMA procedure that "programs the DMA by sending the starting address pointer where ***the PDU*** is located, and its ***total length***." (Ex. 1015.009 (emphasis added).) Thus the DMA procedure, which

Paper 18 (POR) at 38-40; Paper 35 (Reply) at 13.

Dr. Horst (and Dr. Lin) explain that Thia's disclosure is for transmitting data

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORP. and CAVIUM, INC.,
Petitioner,

v.

ALACRITECH, INC.,
Patent Owner.

Case IPR2018-00234[1]
U.S. Patent No. 8,805,948
Title: INTELLIGENT NETWORK INTERFACE SYSTEM AND METHOD FOR PROTOCOL PROCESSING

DECLARATION OF ROBERT HORST, PH.D., IN SUPPORT OF PETITIONER'S REPLY TO PATENT OWNER'S RESPONSE TO PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 8,805,948

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

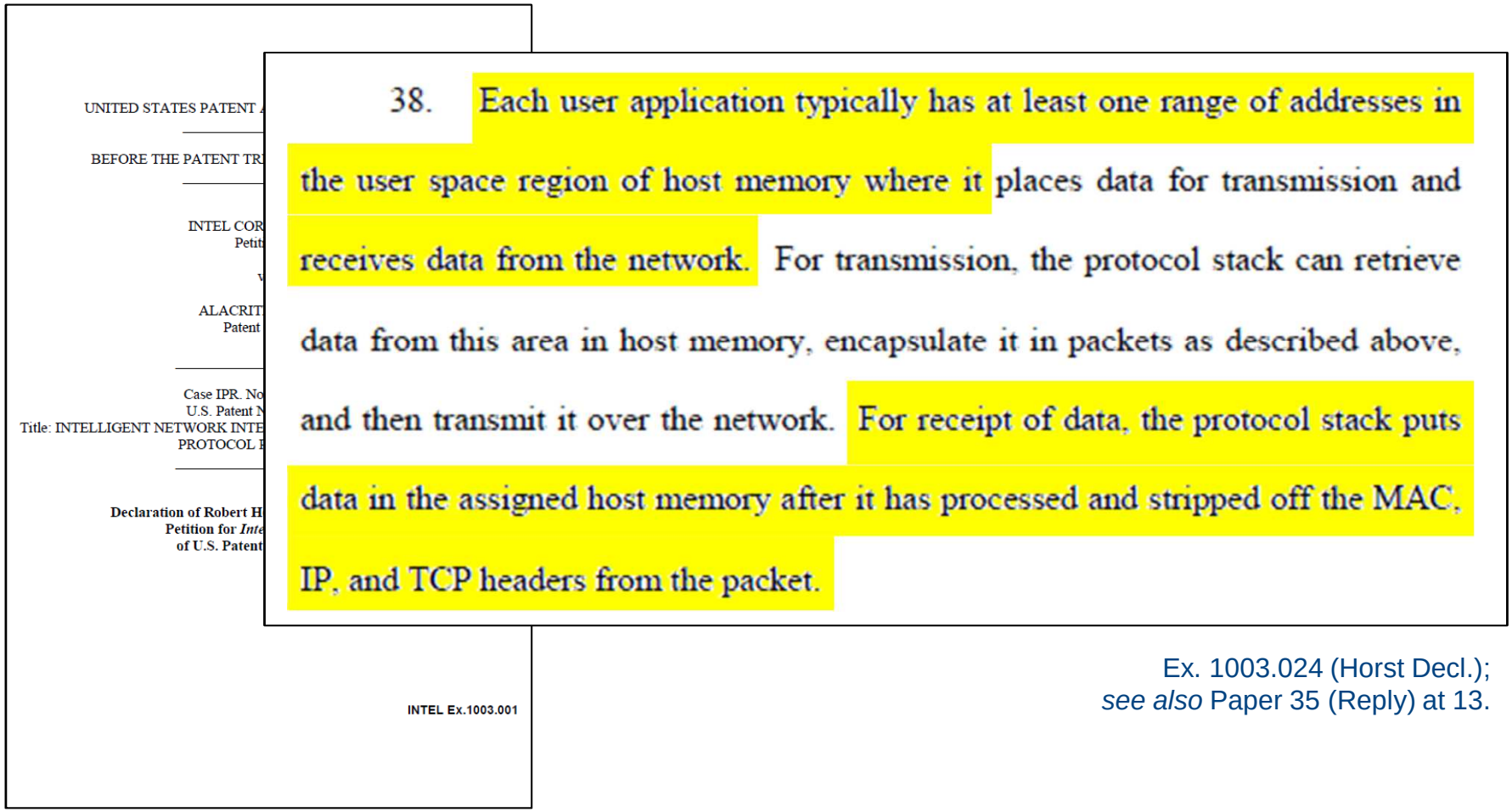
[1] Cavium, Inc., which filed a Petition in Case IPR2018-00403, has been joined as petitioner in this proceeding.

INTEL EX. 1399.00

39. As explained by Dr. Lin in the **1410 Lin Reply Decl. (¶ 30)**, this disclosure is actually describing that the ***host computer*** sends an entire packet (or PDU) to the internal dual ported memory of the ROPE chip for transmission on a network. It does not describe receiving a packet from the network and transferring data from the ROPE chip to the host. Specifically, it states that "the host" is what "initiates the BYPASS_DMA procedure" and "programs the DMA by sending the starting address pointer where the PDU is located …" In other words, the host sends the location of where the PDU is located on the host. "The destination address" for where to send the PDU from the host is "supplied by the bypass chip," and then "DMA transfers the PDU into the internal dual-ported SRAM" of the ROPE chip. *See* Ex. 1015, Thia at .009; *see also id.* at .007, Fig. 2 (illustrating the ROPE chip's "Internal Dual Ported Memory"). Thus, a major premise of Dr. Almeroth's analysis in paragraphs 122 through 128 is false.

Ex. 1399.021-.022 (Horst Reply Decl.);
see also Paper 35 (Reply) at 13.

TCP/IP strips off headers



UNITED STATES PATENT

BEFORE THE PATENT TR

INTEL COR
Petit

ALACRIT
Patent

Case IPR. No
U.S. Patent N
Title: INTELLIGENT NETWORK INTE
PROTOCOL

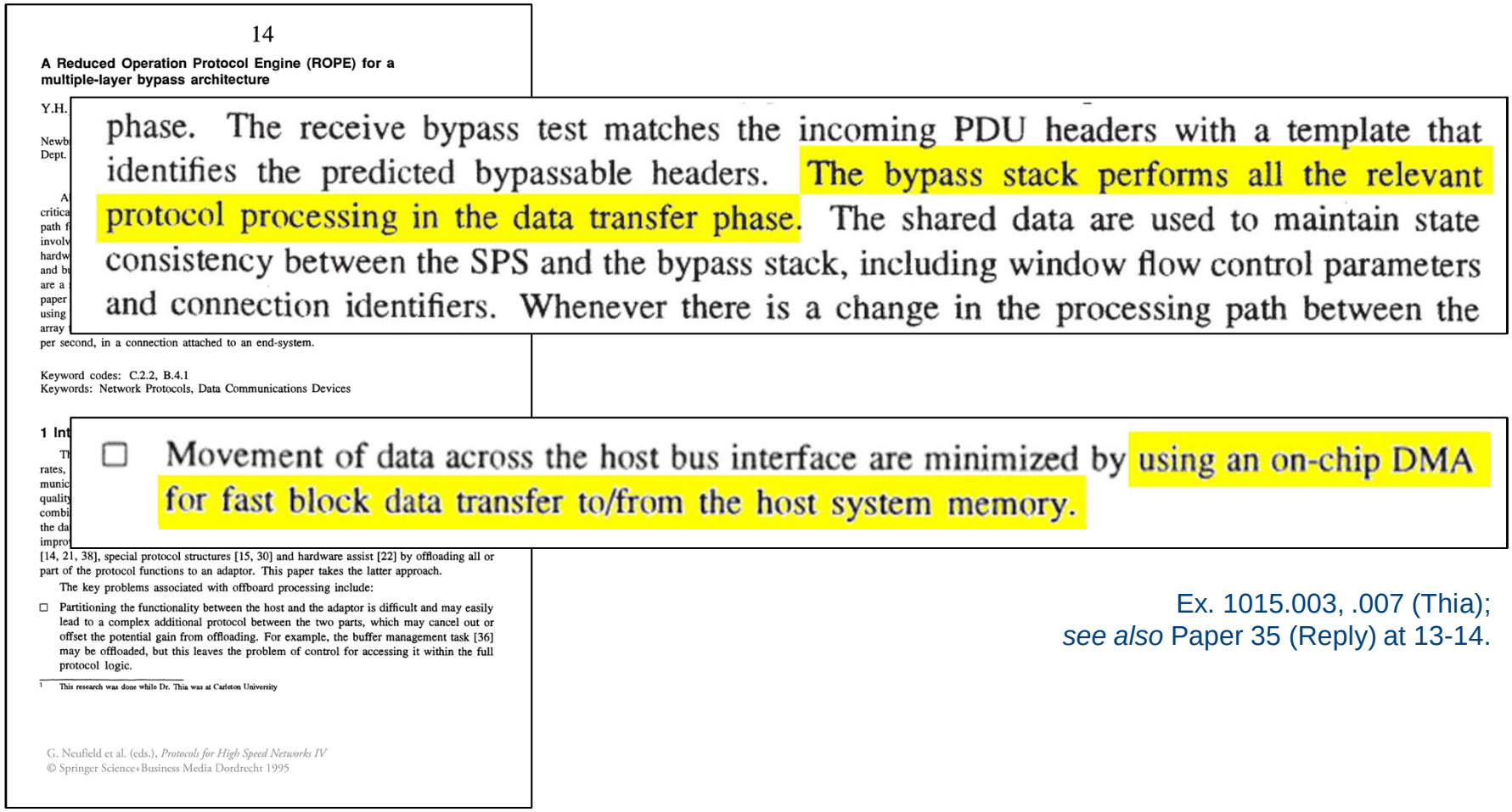
Declaration of Robert H
Petition for *Inte*
of U.S. Patent

INTEL Ex.1003.001

38. Each user application typically has at least one range of addresses in the user space region of host memory where it places data for transmission and receives data from the network. For transmission, the protocol stack can retrieve data from this area in host memory, encapsulate it in packets as described above, and then transmit it over the network. For receipt of data, the protocol stack puts data in the assigned host memory after it has processed and stripped off the MAC, IP, and TCP headers from the packet.

Ex. 1003.024 (Horst Decl.);
see also Paper 35 (Reply) at 13.

Thia discloses transferring data to the host from the ROPE chip after processing the packet



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H.

Newb
Dept.

A
critica
path f
involv
hardw
and b
are a
paper
using
array
per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Int

T
rates,
munic
quality
combi
the da
impro
[14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

phase. The receive bypass test matches the incoming PDU headers with a template that identifies the predicted bypassable headers. The bypass stack performs all the relevant protocol processing in the data transfer phase. The shared data are used to maintain state consistency between the SPS and the bypass stack, including window flow control parameters and connection identifiers. Whenever there is a change in the processing path between the

- Movement of data across the host bus interface are minimized by using an on-chip DMA for fast block data transfer to/from the host system memory.

Ex. 1015.003, .007 (Thia);
see also Paper 35 (Reply) at 13-14.

Thia discloses transferring data to the host from the ROPE chip after processing the packet

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada (**)

Abstract — The Reduced Operation Protocol Engine (ROPE) presented here offloads critical functions of a multiple-layer protocol stack, based on the "bypass concept" of a fast path for data transfer. The motivation for identifying this separate processing path is that it involves only a small subset of the complete protocol, which can then be implemented in hardware. Multiple-layer bypass also eliminates some inter-layer operations such as queue and buffer management, context switching and movement of data across layers, all of which are a significant overhead. ROPE is intended to support high-speed bulk data transfer. The paper describes the design of a ROPE chip for the OSI Session and Transport layer protocols, using VHDL. The design is practical in terms of chip complexity and area, using current gate array technology, and simulation shows that it can support a data rate approaching 1 gigabit per second, in a connection attached to an end-system.

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications Devices

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

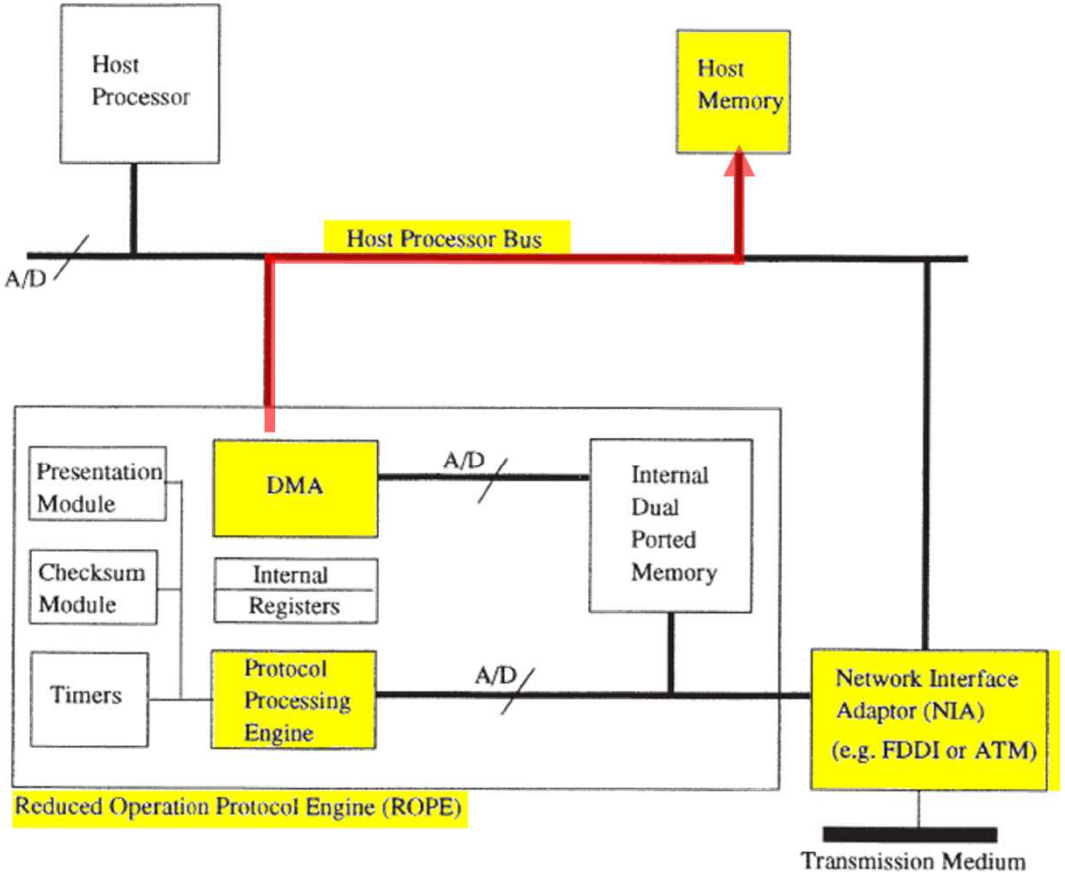
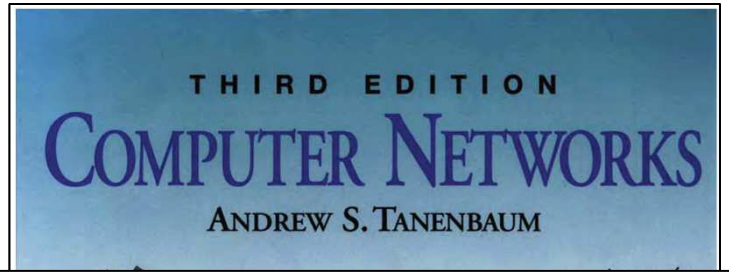


Figure 2 Block Diagram of VLSI bypass system

Ex. 1015.007 (Thia); *see also* Paper 35 (Reply) at 13-14.

Tanenbaum96 teaches that TCP reconstructs the original byte streams (*i.e.* w/o headers)



A TCP entity accepts user data streams from local processes, breaks them up into pieces not exceeding 64K bytes (in practice, usually about 1500 bytes), and sends each piece as a separate IP datagram. When IP datagrams containing TCP data arrive at a machine, they are given to the TCP entity, which reconstructs the original byte streams. For simplicity, we will sometimes use just "TCP" to mean the

Ex.1006.540 (Tanenbaum96);
see also Paper 35 (Reply) at 14.

U.S. Patent No. 7,124,205
(205 Patent)

IPR2018-0226 (Intel)
IPR2018-0400 (Cavium)
IPR2018-1306 (Dell)

*All citations herein are to the IPR2018-00226 case unless otherwise noted.

205 Patent: Instituted Grounds

- **Thia** in view of **SMB**
  - Claims **1**, 4, 5, **8**, 11 and 13
- **Thia** in view of **SMB** and **Carmichael**
  - Claims 6 and 7

Ex. 1015 – Thia, Y.H., Woodside, C.M. Publication (“Thia”)
Ex. 1055 – CAE Specification, Protocols for X/Open PC Interworking: SMB, Version 2 (“SMB”)
Ex. 1053 – U.S. Patent No. 5,894,560 (“Carmichael”)

205 Patent: Disputes

1. Thia is enabling prior art
2. Thia teaches the network interface device performing all network and transport layer processing
3. A POSA would have been motivated to combine Thia and SMB (as well as Carmichael)
4. Motion to Amend 205 Patent should be denied

The Board Has Rejected Many Of PO's Arguments

- This Petition involves same patent and primary reference as in prior related IPRs, including on 205 Patent
- Board has previously rejected PO's arguments
  - 205 FWD at 6-7 – finding Thia teaches network layer bypass (slides 76-82)
  - 205 FWD at 8-9 – finding Thia teaches transport layer bypass (slides 83-90)
  - 205 FWD at 23-24 – finding Thia teaches offloading the full protocol stack, including reassembly, to bypass (slides 83-90)
  - 205 FWD at 9-10 – rejecting PO's argument that Thia as a "feasibility study" undermines motivations to combine (slides 93-94)
  - 205 FWD at 10-14 – rejecting PO's arguments for secondary considerations and finding lack of nexus (slides 190-191)
  - 880 FWD at 8-9 – rejecting PO's arguments that Thia discloses "inoperative device" (slides 68-72)

IPR2017-01405 Paper 84 (205 Patent Final Written Decision)
IPR2017-01409 Paper 79 (880 Patent Final Written Decision).

205 Patent: Disputes

1. **Thia is enabling prior art (Board previously sided with Petitioner)**
2. Thia teaches the network interface device performing all network and transport layer processing
3. A POSA would have been motivated to combine Thia and SMB (as well as Carmichael)
4. Motion to Amend 205 Patent should be denied

PO Fails To Identify Why Thia Is Allegedly Not Enabling

- PO contends that Thia is an "inoperative device" and is therefore a non-enabling reference

Paper 23 (Response) at 18.

- PO's expert, Dr. Almeroth, essentially repeats the opposition and does not provide any additional information or arguments

- A non-enabling reference can be prior art "for all that it teaches"

Id. (citing *Beckman Instruments v. LKB Produkter AB,* 892 F.2d 1547, 1551 (Fed. Cir. 1989)).

Dr. Lin: Thia Is Not A Theoretical Device

4.3 First Design: Design Steps

Figure 3 shows the steps followed in this study. There were three stages, a behavioural model, a structural or RTL model, and a gate level design. These gave us two kinds of feasibility check, that the logic we specified will execute the protocol within the environment we envisage, and that the design is technically feasible, for instance in a reasonable chip area.

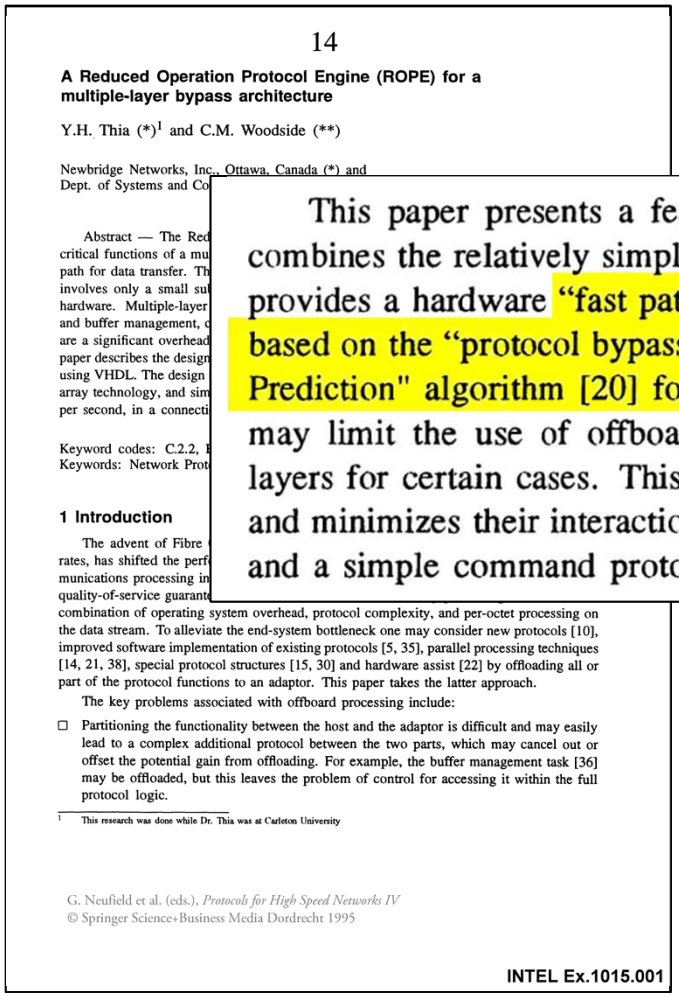
Ex. 1015.008 (Thia).

SYNOPSYS was and still is one of the primary vendors of synthesis design tools used in the semiconductor industry to design semiconductor chips. A POSA would know that a gate-level design can be fabricated into a chip using well-known software tools and chip fabrication facilities. A POSA would have understood the teachings of Thia without the need for Thia to create a final chip.

Ex. 1399, ¶ 7 (Lin Reply Decl.).

- Thia discloses a design ready to be fabricated into a chip

Thia Is Based On Well-known Header Prediction Algorithm



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Co

Abstract — The Red
critical functions of a mu
path for data transfer. Th
involves only a small su
hardware. Multiple-layer
and buffer management, c
are a significant overhead
paper describes the design
using VHDL. The design
array technology, and sim
per second, in a connecti

Keyword codes: C.2.2,
Keywords: Network Prot

1 Introduction

The advent of Fibre
rates, has shifted the perf
munications processing in
quality-of-service guarant
combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

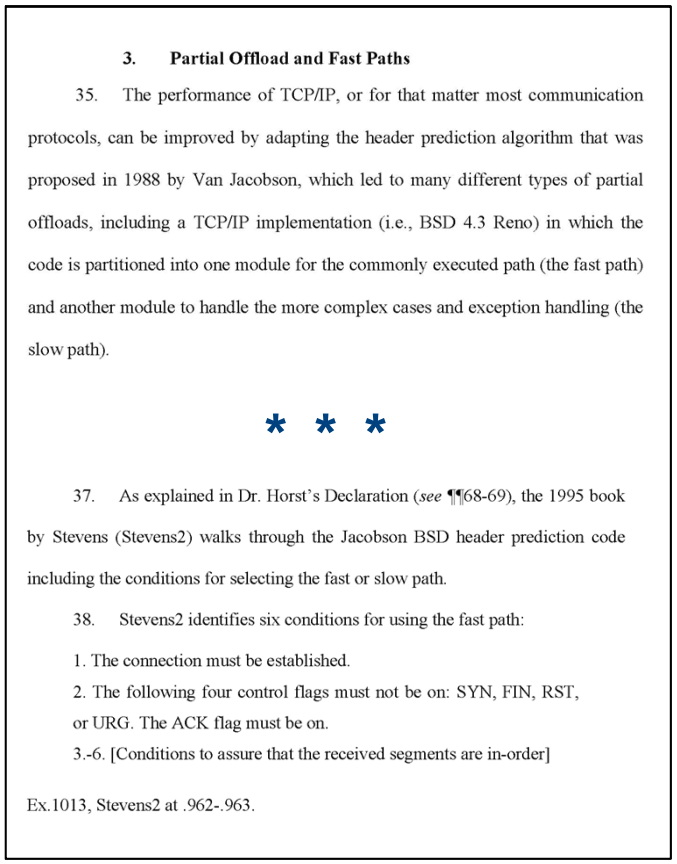
G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

INTEL Ex.1015.001

This paper presents a feasibility study for a new approach to hardware assistance. It combines the relatively simple operations needed for data transfer across multiple layers and provides a hardware "fast path" for them, which will be efficient for bulk data transfer. It is based on the "protocol bypass concept" [37] which is a generalization of Jacobson's "Header Prediction" algorithm [20] for TCP/IP. Bypass solves the problems identified above, which may limit the use of offboard processing, by implementing an entire service through all layers for certain cases. This simplifies the interface between the host and the adaptor chip and minimizes their interaction, which is supported by an access test, some DMA processing and a simple command protocol. The chip design based on bypassing is called ROPE, for

Ex. 1015.002 (Thia); *see also* Paper 1 (Petition) at 24-25.

Dr. Lin: Thia Is Enabling To A POSA



3. Partial Offload and Fast Paths

35. The performance of TCP/IP, or for that matter most communication protocols, can be improved by adapting the header prediction algorithm that was proposed in 1988 by Van Jacobson, which led to many different types of partial offloads, including a TCP/IP implementation (i.e., BSD 4.3 Reno) in which the code is partitioned into one module for the commonly executed path (the fast path) and another module to handle the more complex cases and exception handling (the slow path).

* * *

37. As explained in Dr. Horst's Declaration (*see* ¶¶68-69), the 1995 book by Stevens (Stevens2) walks through the Jacobson BSD header prediction code including the conditions for selecting the fast or slow path.

38. Stevens2 identifies six conditions for using the fast path:

1. The connection must be established.
2. The following four control flags must not be on: SYN, FIN, RST, or URG. The ACK flag must be on.
3.-6. [Conditions to assure that the received segments are in-order]

Ex.1013, Stevens2 at .962-.963.

See Ex. 1003, ¶¶ 35-40 (Lin Decl.);
see also Paper 1 (Petition) at 21, 23.

A POSA would have been able to understand and implement Thia's teachings, which is one of many implementations of Van Jacobson's header prediction

Ex. 1399, ¶¶ 6-7 (Lin Reply Decl.).
see also Ex. 1003, ¶ 71, A-12 – A-14 (Lin Decl.).

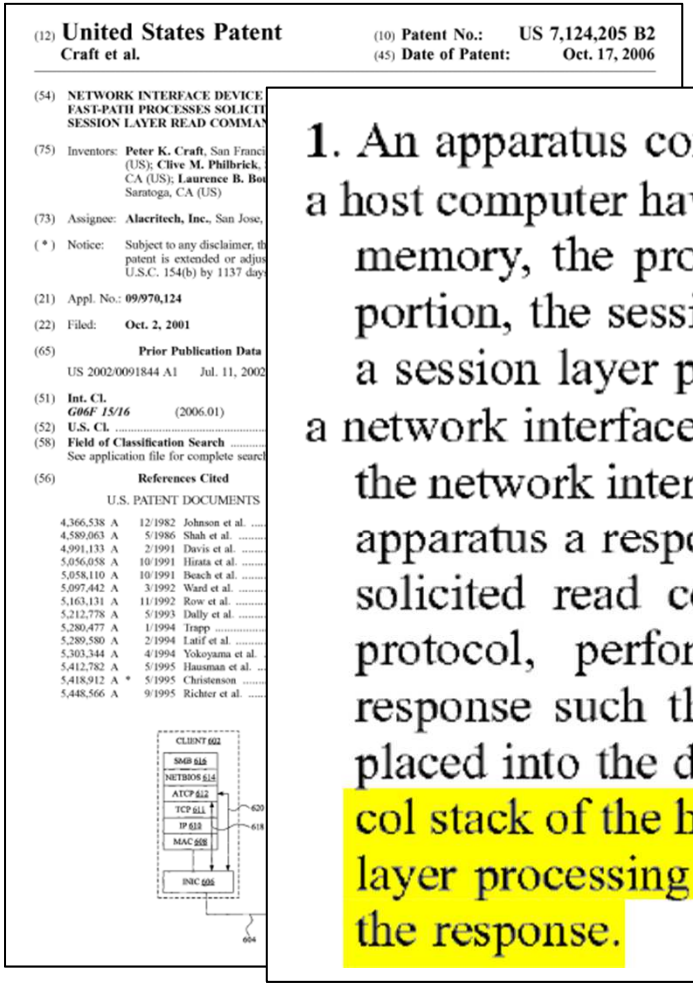
205 Patent: Disputes

2. **Thia teaches the network interface device performing all network and transport layer processing (Board previously sided with Petitioner)**

   a. Thia teaches the network interface device performs all network layer processing

   b. Thia teaches the network interface device performs all transport layer processing

205 Patent: Claim 1



(12) **United States Patent**
Craft et al.

(10) **Patent No.: US 7,124,205 B2**
(45) **Date of Patent: Oct. 17, 2006**

(54) NETWORK INTERFACE DEVICE
FAST-PATH PROCESSES SOLICIT
SESSION LAYER READ COMMA

(75) Inventors: **Peter K. Craft**, San Franci
(US); **Clive M. Philbrick**,
CA (US); **Laurence B. Bo**
Saratoga, CA (US)

(73) Assignee: **Alacritech, Inc.**, San Jose,

(*) Notice: Subject to any disclaimer, th
patent is extended or adjus
U.S.C. 154(b) by 1137 day

(21) Appl. No.: **09/970,124**

(22) Filed: **Oct. 2, 2001**

(65) **Prior Publication Data**

US 2002/0091844 A1 Jul. 11, 2002

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** .....

(58) **Field of Classification Search** .....
See application file for complete searc

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,366,538	A	12/1982	Johnson et al.
4,589,063	A	5/1986	Shah et al.
4,991,133	A	2/1991	Davis et al.
5,056,058	A	10/1991	Hirata et al.
5,058,110	A	10/1991	Beach et al.
5,097,442	A	3/1992	Ward et al.
5,163,131	A	11/1992	Row et al.
5,212,778	A	5/1993	Dally et al.
5,280,477	A	1/1994	Trapp
5,289,580	A	2/1994	Latif et al.
5,303,344	A	4/1994	Yokoyama et al.
5,412,782	A	5/1995	Hausman et al.
5,418,912	A *	5/1995	Christenson
5,448,566	A	9/1995	Richter et al.

1. An apparatus comprising:

a host computer having a protocol stack and a destination memory, the protocol stack including a session layer portion, the session layer portion being for processing a session layer protocol; and

a network interface device coupled to the host computer, the network interface device receiving from outside the apparatus a response to a solicited read command, the solicited read command being of the session layer protocol, performing fast-path processing on the response such that a data portion of the response is placed into the destination memory without the protocol stack of the host computer performing any network layer processing or any transport layer processing on the response.

Ex. 1001 (205 Patent) at Claim 1.

205 Patent: Disputes

2. Thia teaches the network interface device performing all network and transport layer processing (Board previously sided with Petitioner)

   a. **Thia teaches the network interface device performs all network layer processing**

   b. Thia teaches the network interface device performs all transport layer processing

Thia: Bypass All Network Layer Processing In The Data Transfer Phase

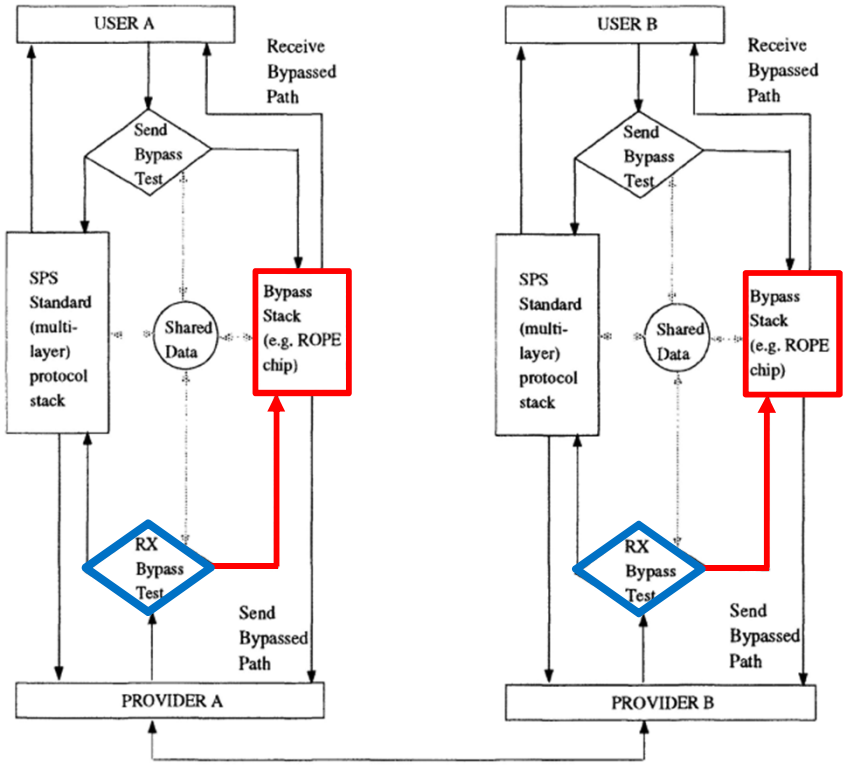


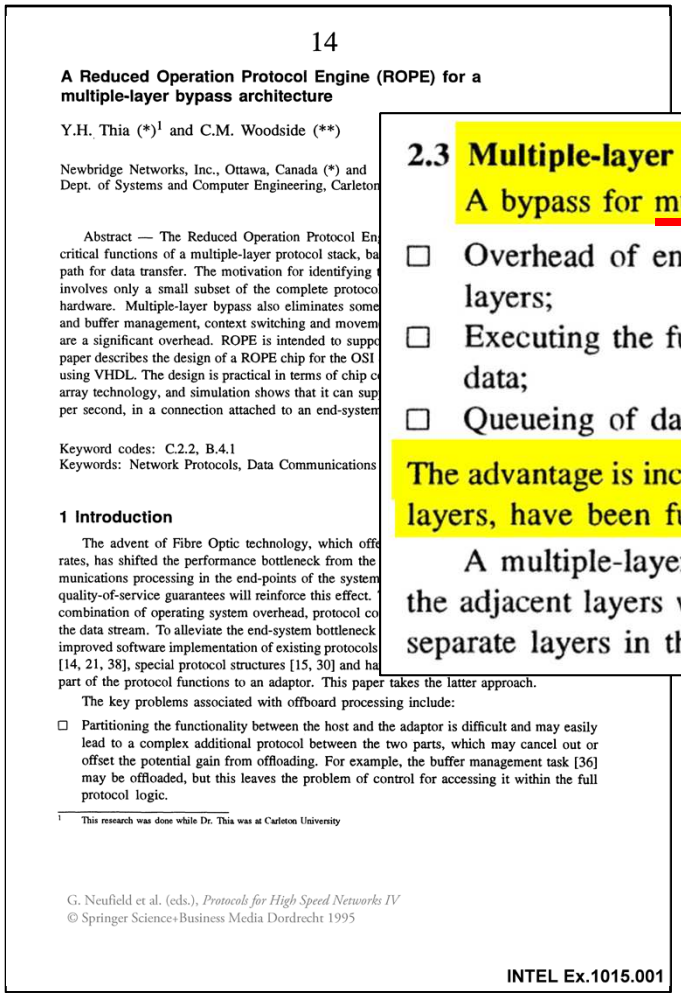
Figure 1 Bypass Architecture

Ex.1015.003 (Thia) at Fig. 1 (annotated);
see, e.g., Ex. 1003, A-14 (Lin Decl.);
see also Paper 1 (Petition) at 49-51.

"The bypass stack performs all the relevant protocol processing in the data transfer phase."

Ex. 1015.003 (Thia);
Paper 1 (Petition) at 50;
Ex. 1003, ¶¶ 74-76 (Lin Decl.);
see also Ex. 1399, ¶ 14 (Lin Reply Decl.).

Thia: Bypass Multiple Layers, Including Network Layer



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and
Dept. of Systems and Computer Engineering, Carleton

Abstract — The Reduced Operation Protocol En
critical functions of a multiple-layer protocol stack, ba
path for data transfer. The motivation for identifying t
involves only a small subset of the complete protoco
hardware. Multiple-layer bypass also eliminates some
and buffer management, context switching and movem
are a significant overhead. ROPE is intended to suppo
paper describes the design of a ROPE chip for the OSI
using VHDL. The design is practical in terms of chip c
array technology, and simulation shows that it can sup
per second, in a connection attached to an end-system

Keyword codes: C.2.2, B.4.1
Keywords: Network Protocols, Data Communications

1 Introduction

The advent of Fibre Optic technology, which offe
rates, has shifted the performance bottleneck from the
munications processing in the end-points of the system
quality-of-service guarantees will reinforce this effect.
combination of operating system overhead, protocol co
the data stream. To alleviate the end-system bottleneck
improved software implementation of existing protocols
[14, 21, 38], special protocol structures [15, 30] and ha
part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

INTEL Ex.1015.001

2.3 Multiple-layer bypass

A bypass for multiple layers instead of just one gives additional gains by avoiding:

- Overhead of encoding and decoding the interface control information passed between layers;
- Executing the full general protocol logic for the layers to decide how to manipulate the data;
- Queueing of data at layer boundaries.

The advantage is increased further in cases where some layers, like the network and application layers, have been further subdivided into sublayers.

A multiple-layer bypass path is a concatenation of processing procedures performed by the adjacent layers when they are simultaneously in the data transfer phase. Meanwhile, the separate layers in the SPS path handle the other phases.

Ex.1015.004 (Thia);
Paper 1 (Petition) at 33, 55, 61;
Paper 42 (Reply) at 6.

OSI Model Has Multiple Layers, Which Must Be Processed In Order

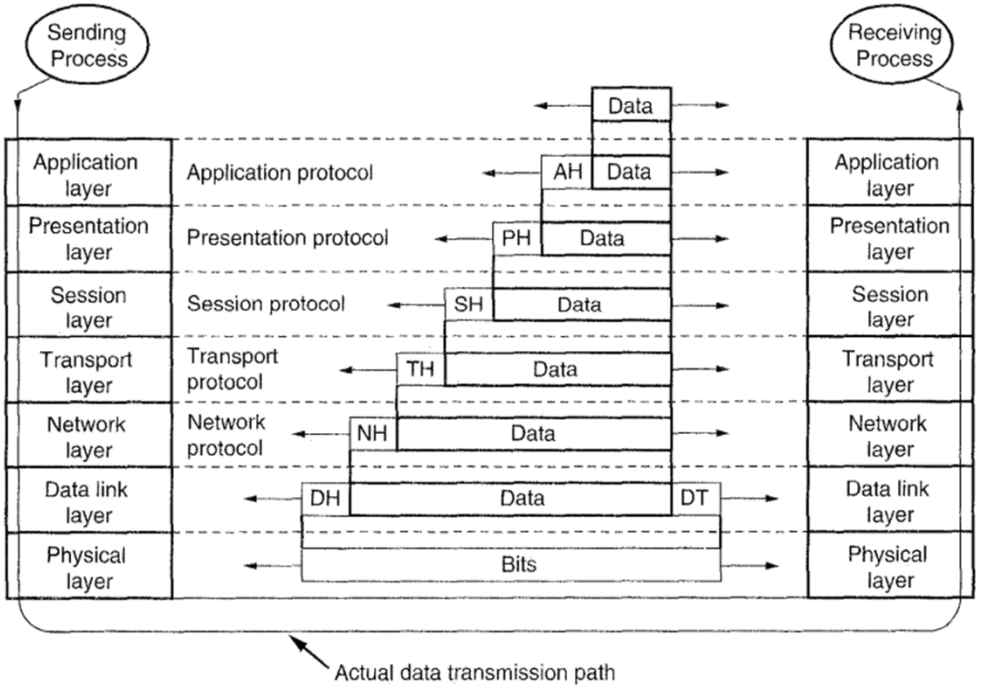


Fig. 1-17. An example of how the OSI model is used. Some of the headers may be null. (Source: H.C. Folts. Used with permission.)

- The network layer must be processed before the transport and session layers
- It is undisputed that Thia discloses processing the transport and session layers on the adapter

See e.g., Ex. 1006 (Tanenbaum96) at Fig. 1-17;
Paper 1 (Petition) at 20-23, 45, 55, 60;
Paper 42 (Reply) at 6-7;
Ex. 1399, ¶¶ 9-11 (Lin Reply Decl.);
Paper 23 (Response) at 2;
Ex. 2026, ¶ 65 (Almeroth Decl.).

205 Patent: Disputes

2. Thia teaches the network interface device performing all network and transport layer processing (Board previously sided with Petitioner)

   a. Thia teaches the network interface device performs all network layer processing

   b. Thia teaches the network interface device performs all transport layer processing

      i. The claims do not recite "reassembly"

      ii. Thia discloses transport layer reassembly of the data portions of packets

      iii. The "segmentation/reassembly" discussed in Thia is below the transport layer

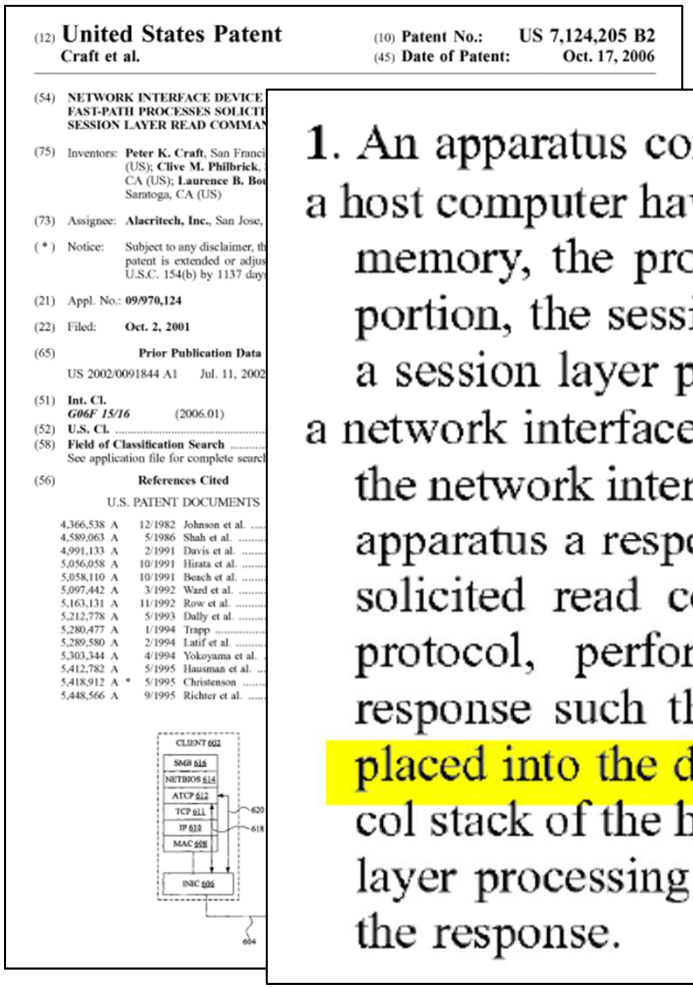
Thia's Transport Layer Bypass Includes "Reassembly"

- PO does not dispute that some transport layer processing is performed on the bypass path, but argues that "reassembly" of incoming packets is missing from Thia:

  "Crucially, Thia does not disclose bypassing the reassembly of incoming packets, which is a primary responsibility of the transport layer"

Paper 23 (Response) at 33-34.

Claims Do Not Recite "Reassembly"



(12) **United States Patent**
Craft et al.

(10) **Patent No.: US 7,124,205 B2**
(45) **Date of Patent: Oct. 17, 2006**

(54) NETWORK INTERFACE DEVICE
FAST-PATH PROCESSES SOLICIT
SESSION LAYER READ COMMA

(75) Inventors: **Peter K. Craft**, San Franci
(US); **Clive M. Philbrick**,
CA (US); **Laurence B. Bo**
Saratoga, CA (US)

(73) Assignee: **Alacritech, Inc.**, San Jose,

(*) Notice: Subject to any disclaimer, th
patent is extended or adjus
U.S.C. 154(b) by 1137 day

(21) Appl. No.: **09/970,124**

(22) Filed: **Oct. 2, 2001**

(65) **Prior Publication Data**

US 2002/0091844 A1 Jul. 11, 2002

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** .....

(58) **Field of Classification Search** .....
See application file for complete searc

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,366,538	A	12/1982	Johnson et al.
4,589,063	A	5/1986	Shah et al.
4,991,133	A	2/1991	Davis et al.
5,056,058	A	10/1991	Hirata et al.
5,058,110	A	10/1991	Beach et al.
5,097,442	A	3/1992	Ward et al.
5,163,131	A	11/1992	Row et al.
5,212,778	A	5/1993	Dally et al.
5,280,477	A	1/1994	Trapp
5,289,580	A	2/1994	Latif et al.
5,303,344	A	4/1994	Yokoyama et al.
5,412,782	A	5/1995	Hausman et al.
5,418,912	A *	5/1995	Christenson
5,448,566	A	9/1995	Richter et al.

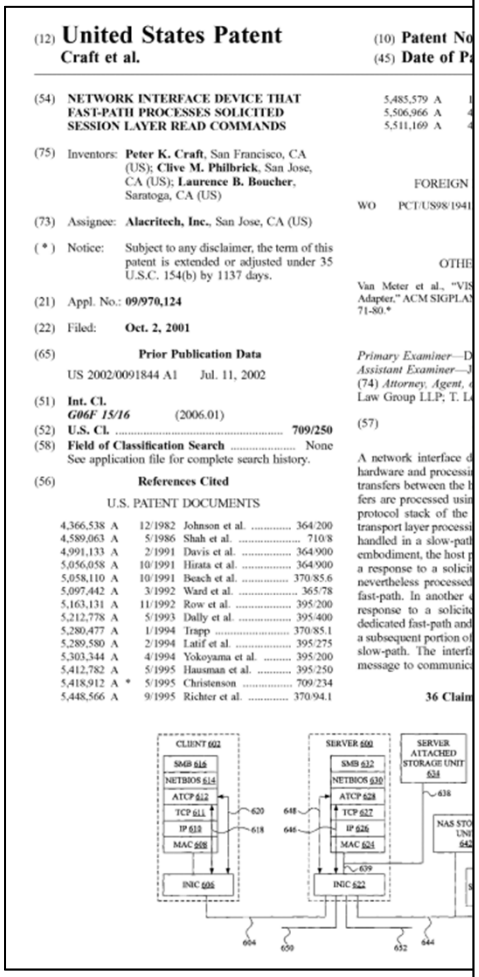
1. An apparatus comprising:

a host computer having a protocol stack and a destination memory, the protocol stack including a session layer portion, the session layer portion being for processing a session layer protocol; and

a network interface device coupled to the host computer, the network interface device receiving from outside the apparatus a response to a solicited read command, the solicited read command being of the session layer protocol, performing fast-path processing on the response such that a data portion of the response is placed into the destination memory without the protocol stack of the host computer performing any network layer processing or any transport layer processing on the response.

Ex. 1001 (205 Patent) at Claim 1.

Claims Do Not Recite "Reassembly"



(12) **United States Patent**
Craft et al.

(10) **Patent No**
(45) **Date of Pa**

(54) **NETWORK INTERFACE DEVICE THAT FAST-PATH PROCESSES SOLICITED SESSION LAYER READ COMMANDS**

(75) Inventors: **Peter K. Craft**, San Francisco, CA (US); **Clive M. Philbrick**, San Jose, CA (US); **Laurence B. Boucher**, Saratoga, CA (US)

(73) Assignee: **Alacritech, Inc.**, San Jose, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1137 days.

(21) Appl. No.: **09/970,124**

(22) Filed: **Oct. 2, 2001**

(65) **Prior Publication Data**

US 2002/0091844 A1 Jul. 11, 2002

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/250**

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,366,538	A	12/1982	Johnson et al.	364/200
4,589,063	A	5/1986	Shah et al.	710/8
4,991,133	A	2/1991	Davis et al.	364/900
5,056,058	A	10/1991	Hirata et al.	364/900
5,058,110	A	10/1991	Beach et al.	370/85.6
5,097,442	A	3/1992	Ward et al.	365/78
5,163,131	A	11/1992	Row et al.	395/200
5,212,778	A	5/1993	Dally et al.	395/400
5,280,477	A	1/1994	Trapp	370/85.1
5,289,580	A	2/1994	Latif et al.	395/275
5,303,344	A	4/1994	Yokoyama et al.	395/200
5,412,782	A	5/1995	Hausman et al.	395/250
5,418,912	A *	5/1995	Christenson	709/234
5,448,566	A	9/1995	Richter et al.	370/94.1

5,485,579	A		
5,506,966	A		
5,511,169	A		

FOREIGN

WO PCT/US98/1941

OTHE

Van Meter et al., "VIS Adapter," ACM SIGPLA 71-80.*

Primary Examiner—D
Assistant Examiner—J
(74) *Attorney, Agent,* Law Group LLP; T. L

(57)

A network interface d hardware and processi transfers between the h fers are processed usi protocol stack of the transport layer processi handled in a slow-path embodiment, the host p a response to a solici nevertheless processed fast-path. In another response to a solicit dedicated fast-path and a subsequent portion o slow-path. The interf message to communic

36 Claim

8. A method, comprising:

issuing a read request to a network storage device, the read request passing through a network to the network storage device;

receiving on a network interface device a packet from the network storage device in response to the read request, the packet including data, the network interface device being coupled to a host computer by a bus, the host computer having a protocol stack for carrying out network layer and transport layer processing;

performing fast-path processing on the packet such that the data is placed into a destination memory without the protocol stack of the host computer doing any network layer processing on the packet and without the protocol stack of the host computer doing any transport layer processing on the packet;

receiving on the network interface device a subsequent packet from the network storage device in response to the read request, the subsequent packet including subsequent data; and

performing slow-path processing on the subsequent packet such that the protocol stack of the host computer does network layer processing and transport layer processing on the subsequent packet.

Ex. 1001 (205 Patent) at Claim 8.

205 Patent: Disputes

2. Thia teaches the network interface device performing all network and transport layer processing (Board previously sided with Petitioner)

   a. Thia teaches the network interface device performs all network layer processing

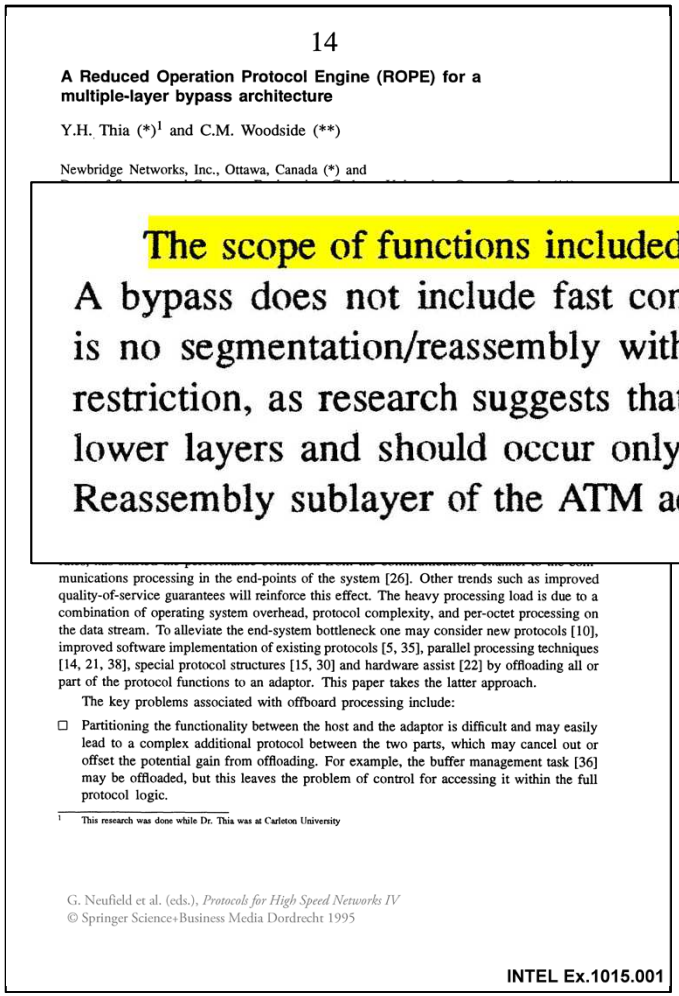
   b. Thia teaches the network interface device performs all transport layer processing

      i. The claims do not recite "reassembly"

      ii. Thia discloses transport layer reassembly of the data portions of packets

      iii. The "segmentation/reassembly" discussed in Thia is below the transport layer

Thia: Bypass Functions Can Be Extended



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and

munications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufield et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

INTEL Ex.1015.001

The scope of functions included in a bypass may be narrowly defined, or more extended. A bypass does not include fast connection setup but also does not interfere with it. There is no segmentation/reassembly within the bypass path, but we do not see this as a major restriction, as research suggests that fragmentation of PDUs should be restricted only to the lower layers and should occur only once in the protocol stack [23]. The Segmentation and Reassembly sublayer of the ATM adaptation layer is a good place for such functions [25].

Ex. 1015.014 (Thia);
Ex. 1399, ¶ 16 (Lin Reply Decl.).

Thia: Put Incoming Packets In The Right Order In The Transport Layer

4.5 Second Design, including major procedures for Transport Class 4 (Implemented)

This section describes extensions to the first design, which only supports Session BCS and TP2 functionality, to include some common TP4 functionality. Procedures for checksum, retransmission on timeout and resequencing were implemented. Extensions to the Session layer functionality and procedures for presentation layer conversion were not implemented, but are also discussed in section 6.

4.5.3 Retransmission and Resequencing

At the receiver end, out-of-sequence PDUs outside the flow-control window will be discarded. Otherwise, a PDU is buffered for resequencing. Duplicate TPDUs can be detected

Ex.1015.010 (Thia);
Paper 42 (Reply) at 9-10.

Thia: DMA Data Portions Of PDUs To The Host In The Bypass Path

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and

This places the maximum stress on the ROPE chip.The architectural considerations involved in the chip design can be summarized as follows:

- Movement of data across the host bus interface are minimized by using an on-chip DMA for fast block data transfer to/from the host system memory.

1 Introduction

The advent of Fibre Optic technology, which offers high bandwidth and low bit error rates, has shifted the performance bottleneck from the communications channel to the communications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

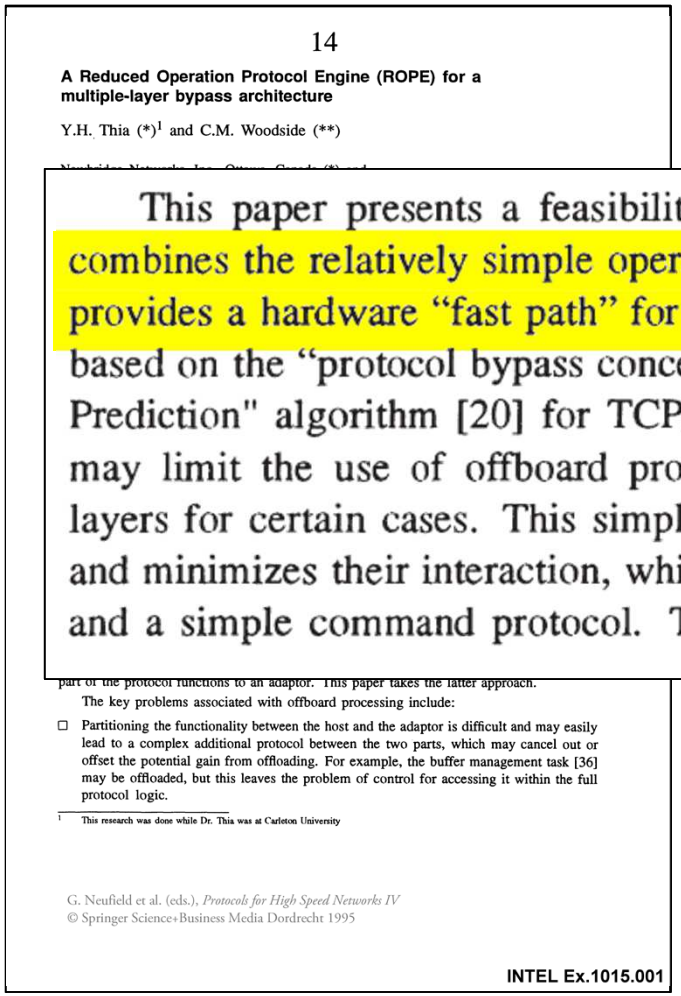
[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

INTEL Ex.1015.001

Ex. 1015.007 (Thia);
Paper 42 (Reply) at 9-10;
Ex. 1399 (Lin Reply Decl.) ¶ 17.

Thia: DMA Data Portions Of PDUs To The Host In The Bypass Path



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

This paper presents a feasibility study for a new approach to hardware assistance. It combines the relatively simple operations needed for data transfer across multiple layers and provides a hardware "fast path" for them, which will be efficient for bulk data transfer. It is based on the "protocol bypass concept" [37] which is a generalization of Jacobson's "Header Prediction" algorithm [20] for TCP/IP. Bypass solves the problems identified above, which may limit the use of offboard processing, by implementing an entire service through all layers for certain cases. This simplifies the interface between the host and the adaptor chip and minimizes their interaction, which is supported by an access test, some DMA processing and a simple command protocol. The chip design based on bypassing is called ROPE, for

part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

INTEL Ex.1015.001

Ex. 1015.002 (Thia);
Ex. 1399 (Lin Reply Decl.) ¶ 17.

205 Patent: Disputes

2. Thia teaches the network interface device performing all network and transport layer processing (Board previously sided with Petitioner)

   a. Thia teaches the network interface device performs all network layer processing

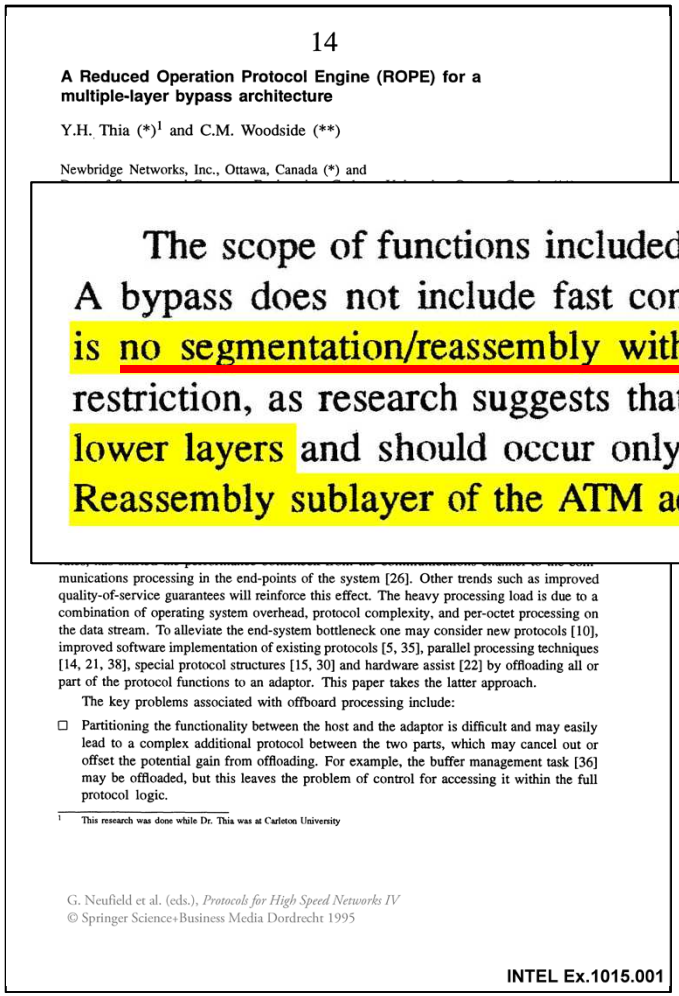
   b. Thia teaches the network interface device performs all transport layer processing

      i. The claims do not recite "reassembly"

      ii. Thia discloses transport layer reassembly of the data portions of packets

      iii. The "segmentation/reassembly" discussed in Thia is below the transport layer

Thia's Segmentation/Reassembly For ATM Is Not Transport Layer Reassembly



14

A Reduced Operation Protocol Engine (ROPE) for a multiple-layer bypass architecture

Y.H. Thia (*)[1] and C.M. Woodside (**)

Newbridge Networks, Inc., Ottawa, Canada (*) and

munications processing in the end-points of the system [26]. Other trends such as improved quality-of-service guarantees will reinforce this effect. The heavy processing load is due to a combination of operating system overhead, protocol complexity, and per-octet processing on the data stream. To alleviate the end-system bottleneck one may consider new protocols [10], improved software implementation of existing protocols [5, 35], parallel processing techniques [14, 21, 38], special protocol structures [15, 30] and hardware assist [22] by offloading all or part of the protocol functions to an adaptor. This paper takes the latter approach.

The key problems associated with offboard processing include:

- Partitioning the functionality between the host and the adaptor is difficult and may easily lead to a complex additional protocol between the two parts, which may cancel out or offset the potential gain from offloading. For example, the buffer management task [36] may be offloaded, but this leaves the problem of control for accessing it within the full protocol logic.

[1] This research was done while Dr. Thia was at Carleton University

G. Neufeld et al. (eds.), *Protocols for High Speed Networks IV*
© Springer Science+Business Media Dordrecht 1995

INTEL Ex.1015.001

The scope of functions included in a bypass may be narrowly defined, or more extended. A bypass does not include fast connection setup but also does not interfere with it. There is no segmentation/reassembly within the bypass path, but we do not see this as a major restriction, as research suggests that fragmentation of PDUs should be restricted only to the lower layers and should occur only once in the protocol stack [23]. The Segmentation and Reassembly sublayer of the ATM adaptation layer is a good place for such functions [25].

Ex. 1015.014 (Thia).

Thia's "segmentation/reassembly" is fragmenting/re-assembling portions of packets at a layer below the transport layer.

See, e.g., Paper 42 (Reply) at 8-9;
Ex. 1399 (Lin Reply Decl.) ¶ 15.

Dr. Lin: Thia's Segmentation/Reassembly For ATM Not Transport Layer Reassembly

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DECLARATIO
TO PATENT O

Mail Stop "PATE
Patent Trial and A
U.S. Patent and Tr
P.O. Box 1450
Alexandria, VA 2

layers, such as the transport layer. Thia's disclosure of "no segmentation/reassembly within the bypass path" is addressing this lower layer segmentation/re-assembly. This is confirmed by Thia's statement that "fragmentation of PDUs should be restricted only to the lower layers and should occur only once in the protocol stack…." Ex. 1015.014. In fact, the same sentence in Thia stating "[t]here is no segmentation/reassembly in the bypass path" ends with a citation to a paper that is addressing IP fragmentation. *See* Ex. 1218.001

[1] Cavium, Inc. ("Cavium") which filed a Petition for Inter Partes Review in Case No. IPR2018-0400, has been joined as a petitioner in this proceeding.

INTEL EX. 1399

Ex. 1399 (Lin Reply Decl.) ¶ 15.

205 Patent: Disputes

1. Thia is enabling prior art
2. Thia teaches the network interface device performing all network and transport layer processing
3. **A POSA would have been motivated to combine Thia and SMB (as well as Carmichael)**
4. Motion to Amend 205 Patent should be denied

205 Patent: Disputes

3. A POSA would have been motivated to combine Thia and SMB (as well as Carmichael)
   a. **A POSA would have used Thia's bypass system with the SMB protocol of the SMB reference**
   b. The motivations to further include Carmichael are unrebutted by PO
   c. The Petition includes sufficient evidence regarding expectation of success

Thia's Bypass Would Have Been Improved By SMB's SMB Protocol

Third, it would have obvious to combine Thia with SMB to improve Thia by adding the functionality provided by SMB. Namely, SMB provides file-sharing and print-sharing services (among several other services). *See, e.g.*, Ex.1055, SMB at .022. Such services are demanded in network environments and a POSA would have been motivated to adapt Thia to include such a communications protocol (such as SMB). Ex.1003, Lin Decl. ¶92. SMB also provides network communications, which makes it especially appropriate for Thia because Thia is designed to work in a network environment. *See* Ex.1015, Thia at Fig. 2 (having a "Network Interface Adaptor"); Ex.1055, SMB at .032. SMB also provides security services, which are always attractive in a network environment. *Id.* at .033; Ex.1003, Lin Decl. ¶92.

Paper 1 (Petition) at 40;
see also Ex.1003 (Lin Decl.) ¶¶ 90-95.

PO's Only Criticism Of Combining Thia And SMB Is Thia Is Theoretical Reference

> Petitioners' proffered "Motivations To Combine *Thia* and SMB" ***all*** erroneously assume that *Thia* discloses a real-world chip, and purport to offer motivations to combine that supposed real-world chip with *X/Open SMB* in a real-world environment. (*See* Intel Petition §11.1; Cavium Petition §11.1; Ex. 2026, Almeroth Decl. ¶102.) Because *Thia* actually only discloses a feasibility study using a theoretical chip in a simulated environment, those proffered motivations fail. (Ex. 2026, Almeroth Decl. ¶102.)

Paper 23 (Response) at 36-37.

- Board previously rejected this argument

IPR2017-01405 Paper 84 (205 Patent Final Written Decision).

205 Patent: Disputes

3. A POSA would have been motivated to combine Thia and SMB (as well as Carmichael)
   a. A POSA would have used Thia's bypass system with the SMB protocol of the SMB reference
   b. <u>The motivations to further include Carmichael are unrebutted by PO</u>
   c. The Petition includes sufficient evidence regarding expectation of success

A POSA Would Have Been Motivated To Further Include Carmichael

- PO does not address motivations to further include Carmichael

Paper 23 (Response) at 40-41.

205 Patent: Disputes

3. A POSA would have been motivated to combine Thia and SMB (as well as Carmichael)
   a. A POSA would have used Thia's bypass system with the SMB protocol of the SMB reference
   b. The motivations to further include Carmichael are unrebutted by PO
   c. <u>The Petition includes sufficient evidence regarding expectation of success</u>

Dr. Lin: Reasonable Expectation Of Successful Combination

94. A person of ordinary skill would have also recognized that the combination of Thia and SMB would be fairly easy to implement and would have a predictable result. Specifically, the combination would provide the beneficial fast path system to SMB commands. Both Thia and SMB disclose that they are applicable to the OSI model. Ex.1055, SMB at .002; Ex.1015, Thia at Abstract

Title: N

DECLARATION OF BILL LIN IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,124,205 UNDER 37 C.F.R. § 1.68

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

INTEL Ex.1003.001

Ex. 1003 (Lin Decl.) ¶ 94; *see also* Paper 1 (Petition) at 40-41.

PO Failed To Rebut Reasonable Expectation Of Successful Combination

- PO failed to identify any reason why there would not be a reasonable expectation of success
- Did not provide any expert testimony or evidence to the contrary

See Paper 23 (Response) at 26-27.

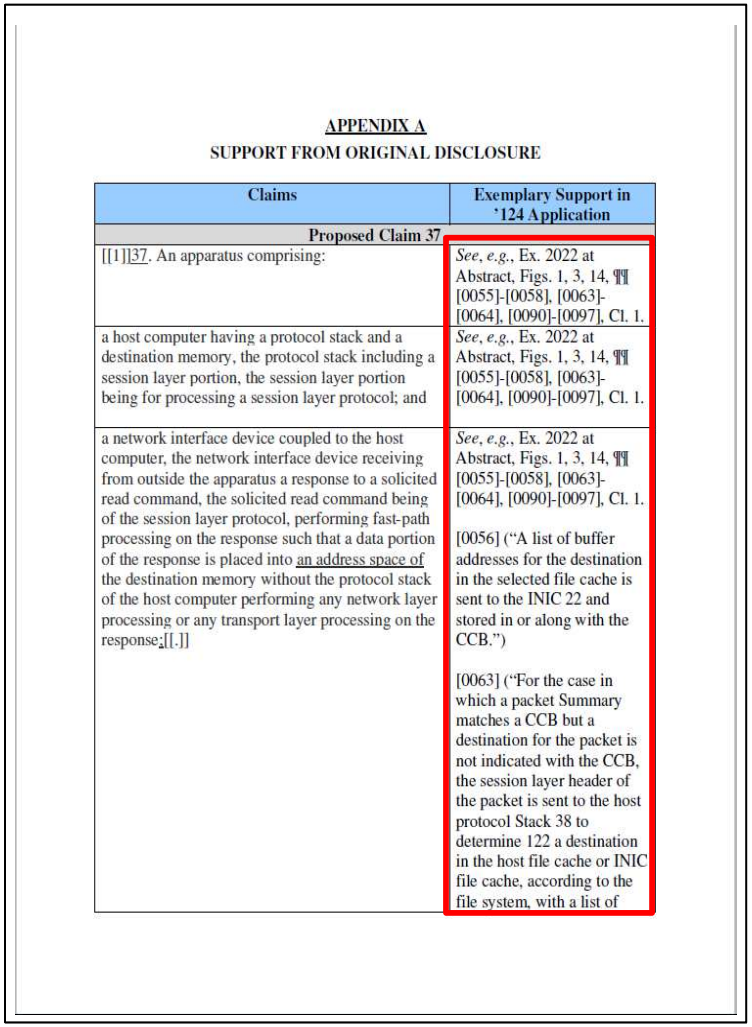
205 Patent: Disputes

4. Motion to Amend 205 Patent should be denied

   a. **PO has not met its burden of production under 35 U.S.C. § 316(d) due to its failure to provide adequate written description support**

   b. The prior art discloses each limitation of the substitute claims

PO Provided Identical String Citations For All Limitations

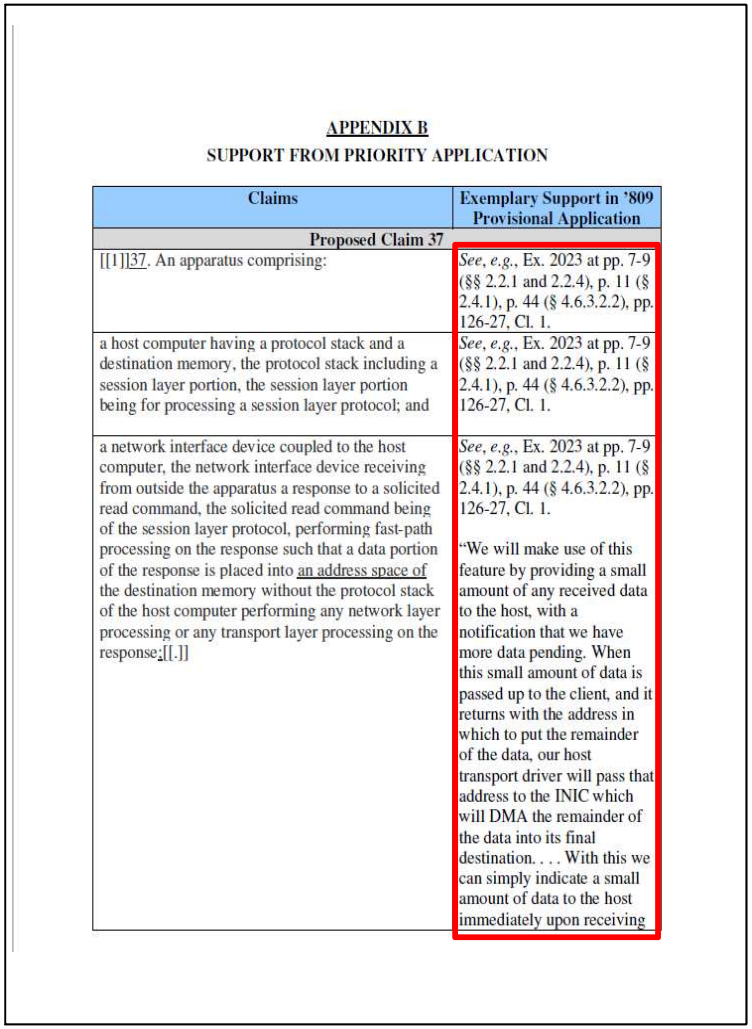


APPENDIX A

SUPPORT FROM ORIGINAL DISCLOSURE

Claims	Exemplary Support in '124 Application
Proposed Claim 37	
[[1]]37. An apparatus comprising:	*See, e.g.*, Ex. 2022 at Abstract, Figs. 1, 3, 14, ¶¶ [0055]-[0058], [0063]-[0064], [0090]-[0097], Cl. 1.
a host computer having a protocol stack and a destination memory, the protocol stack including a session layer portion, the session layer portion being for processing a session layer protocol; and	*See, e.g.*, Ex. 2022 at Abstract, Figs. 1, 3, 14, ¶¶ [0055]-[0058], [0063]-[0064], [0090]-[0097], Cl. 1.
a network interface device coupled to the host computer, the network interface device receiving from outside the apparatus a response to a solicited read command, the solicited read command being of the session layer protocol, performing fast-path processing on the response such that a data portion of the response is placed into an address space of the destination memory without the protocol stack of the host computer performing any network layer processing or any transport layer processing on the response;[[.]]	*See, e.g.*, Ex. 2022 at Abstract, Figs. 1, 3, 14, ¶¶ [0055]-[0058], [0063]-[0064], [0090]-[0097], Cl. 1. [0056] ("A list of buffer addresses for the destination in the selected file cache is sent to the INIC 22 and stored in or along with the CCB.") [0063] ("For the case in which a packet Summary matches a CCB but a destination for the packet is not indicated with the CCB, the session layer header of the packet is sent to the host protocol Stack 38 to determine 122 a destination in the host file cache or INIC file cache, according to the file system, with a list of

Paper 20 (Motion to Amend) at Appendix A, 8.



APPENDIX B

SUPPORT FROM PRIORITY APPLICATION

Claims	Exemplary Support in '809 Provisional Application
Proposed Claim 37	
[[1]]37. An apparatus comprising:	*See, e.g.*, Ex. 2023 at pp. 7-9 (§§ 2.2.1 and 2.2.4), p. 11 (§ 2.4.1), p. 44 (§ 4.6.3.2.2), pp. 126-27, Cl. 1.
a host computer having a protocol stack and a destination memory, the protocol stack including a session layer portion, the session layer portion being for processing a session layer protocol; and	*See, e.g.*, Ex. 2023 at pp. 7-9 (§§ 2.2.1 and 2.2.4), p. 11 (§ 2.4.1), p. 44 (§ 4.6.3.2.2), pp. 126-27, Cl. 1.
a network interface device coupled to the host computer, the network interface device receiving from outside the apparatus a response to a solicited read command, the solicited read command being of the session layer protocol, performing fast-path processing on the response such that a data portion of the response is placed into an address space of the destination memory without the protocol stack of the host computer performing any network layer processing or any transport layer processing on the response;[[.]]	*See, e.g.*, Ex. 2023 at pp. 7-9 (§§ 2.2.1 and 2.2.4), p. 11 (§ 2.4.1), p. 44 (§ 4.6.3.2.2), pp. 126-27, Cl. 1. "We will make use of this feature by providing a small amount of any received data to the host, with a notification that we have more data pending. When this small amount of data is passed up to the client, and it returns with the address in which to put the remainder of the data, our host transport driver will pass that address to the INIC which will DMA the remainder of the data into its final destination. . . . With this we can simply indicate a small amount of data to the host immediately upon receiving

Paper 20 (Motion to Amend) at Appendix B, 19.

PO's Amendments Lack Written Description Support

- PO provides no explanation for why the alleged written description supports "allocating the [first] address space of the destination memory for placement of data" in substitute claim 37 or 42
- Only specific quote in support is from Paragraph 56 of the 124 Application
- Paragraph 56 is about processing a file write message, not a response to a solicited read command

PO's Amendments Lack Written Description Support

- "Address space" used for first time in substitute claims
- Not in specification or original claims
- No claim construction offered
- Definition not clear to a POSA

> 27. It is unclear what Patent Owner is referring to with the term "address space" because the claims do not use the term as one of ordinary skill in the art would use the term. A person of ordinary skill in the art would refer to allocating blocks of memory within an existing address space. In contrast, the claims refer to "allocating the address space of the destination memory for placement of data."

See Ex.1305 (Lin Decl. ISO Petitioner's Opp. To Mtn. to Amend) ¶¶ 25-27.

205 Patent: Disputes

4. Motion to Amend 205 Patent should be denied

a. PO has not met its burden of production under 35 U.S.C. § 316(d) due to its failure to provide adequate written description support

b. <u>The prior art discloses each limitation of the substitute claims</u>

205 Patent: Grounds For Substitute Claims

- **Thia** in combination with **SMB** and **APA**
  - Claims **37**-39 and **42**-44
- **Thia** in combination with **SMB, Carmichael,** and **APA**
  - Claims 40-41

1997 Provisional's Teachings Of Windows NT Are Admitted Prior Art (APA)

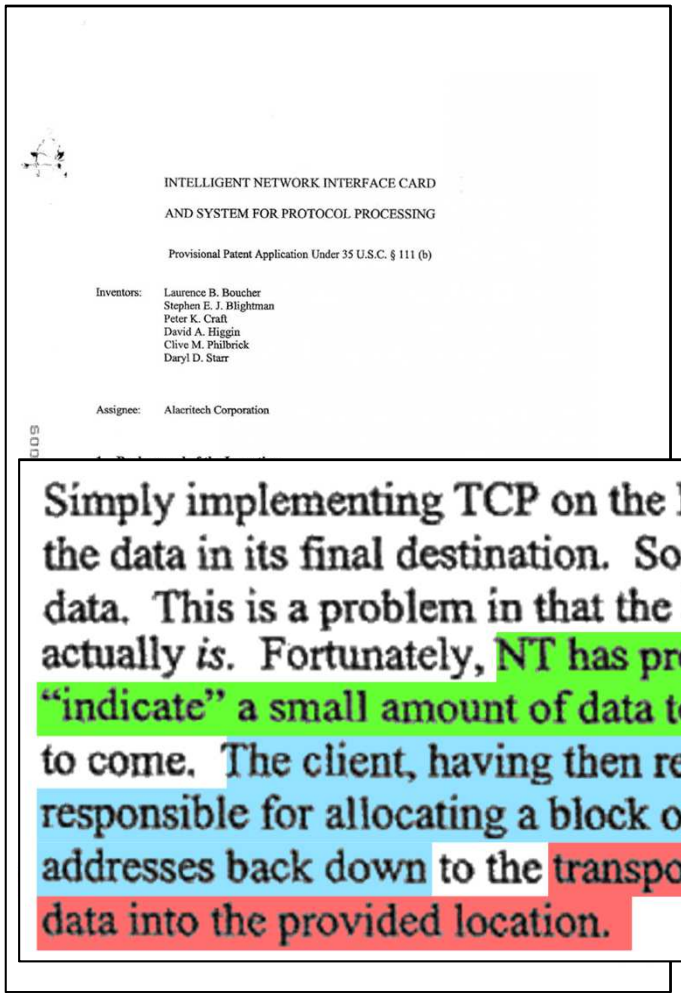
- "A statement in a patent that something is in the prior art is binding on the applicant and patentee for determinations of anticipation and obviousness."

  WesternGeco LLC v. ION Geophysical Corp., 889 F.3d 1308, 1329-30 (Fed. Cir. 2018).

- Admitted prior art falls within 35 U.S.C. § 311(b) and "a patentee's admissions constitute background knowledge that may be imputed to a person of ordinary skill in the art for purposes of an obviousness analysis."

  G.B.T. Inc. v. Walletex Microelectronics Ltd., IPR2018-00326, Paper 14 at 15 (P.T.A.B. Jul. 5, 2018) (citing *Randall Mfg. v. Rea,* 733 F.3d 1355, 1363 (Fed. Cir. 2013).

1997 Provisional Admits The Features In Its Amendments Are In The Prior Art



INTELLIGENT NETWORK INTERFACE CARD

AND SYSTEM FOR PROTOCOL PROCESSING

Provisional Patent Application Under 35 U.S.C. § 111 (b)

Inventors: Laurence B. Boucher
Stephen E. J. Blightman
Peter K. Craft
David A. Higgin
Clive M. Philbrick
Daryl D. Starr

Assignee: Alacritech Corporation

Claim Language:

"wherein the fast-path processing of the response follows the protocol stack processing a first response to the solicited read command to set up a fast-path connection"

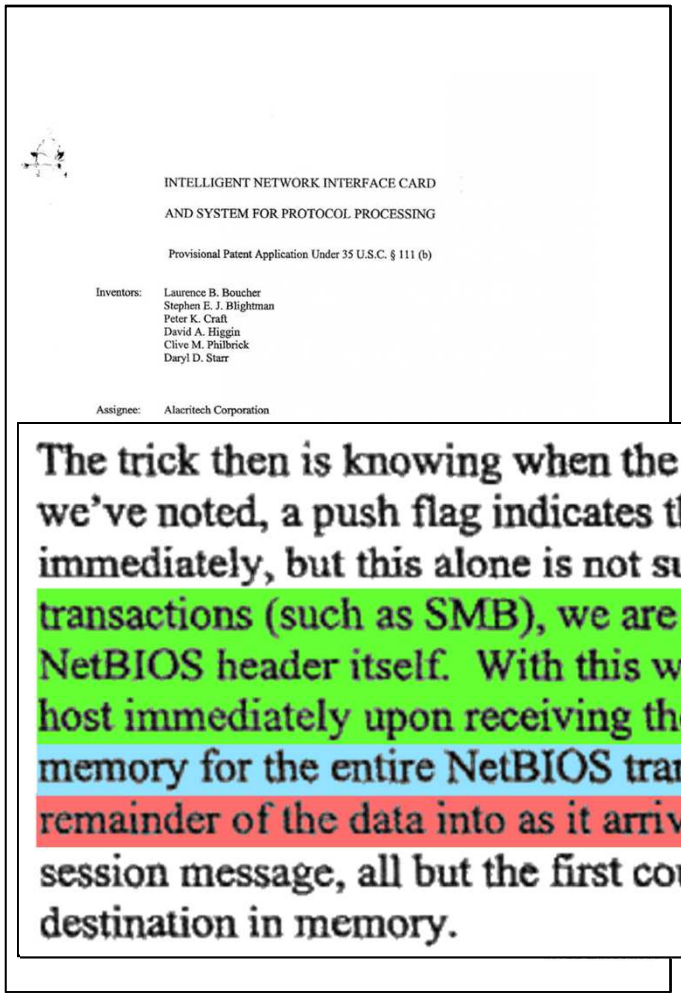
"allocating the address space of the destination memory for placement of data"

"placing a data portion of the first response into the address of the destination memory"

Simply implementing TCP on the INIC does not allow us to achieve our goal of landing the data in its final destination. Somehow the host has to tell the INIC where to put the data. This is a problem in that the host can not do this without knowing what the data actually *is*. Fortunately, NT has provided a mechanism by which a transport driver can "indicate" a small amount of data to a client above it while telling it that it has more data to come. The client, having then received enough of the data to know what it is, is then responsible for allocating a block of memory and passing the memory address or addresses back down to the transport driver, which is in turn responsible for moving the data into the provided location.

Ex. 1031 (1997 Provisional FH) at .011-.012.

1997 Provisional Admits The Features In Its Amendments Are In The Prior Art



INTELLIGENT NETWORK INTERFACE CARD

AND SYSTEM FOR PROTOCOL PROCESSING

Provisional Patent Application Under 35 U.S.C. § 111 (b)

Inventors: Laurence B. Boucher
Stephen E. J. Blightman
Peter K. Craft
David A. Higgin
Clive M. Philbrick
Daryl D. Starr

Assignee: Alacritech Corporation

Claim Language:

"wherein the fast-path processing of the response follows the protocol stack processing a first response to the solicited read command to set up a fast-path connection"

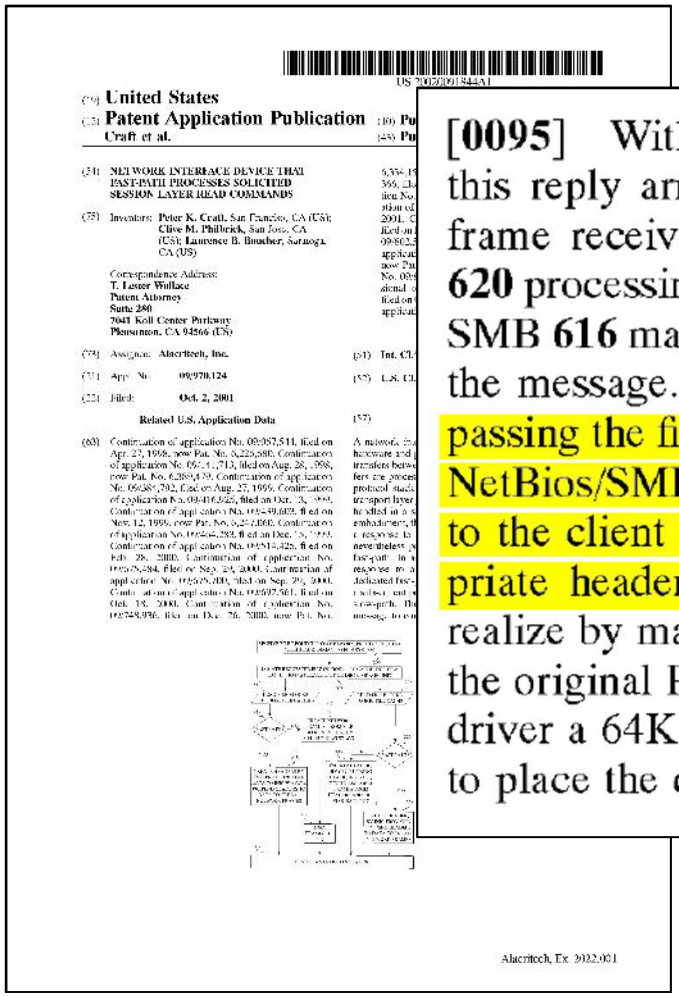
"allocating the address space of the destination memory for placement of data"

"placing a data portion of the first response into the address of the destination memory"

The trick then is knowing when the data should be delivered to the client or not. As we've noted, a push flag indicates that the data should be delivered to the client immediately, but this alone is not sufficient. Fortunately, in the case of NetBIOS transactions (such as SMB), we are explicitly told the length of the session message in the NetBIOS header itself. With this we can simply indicate a small amount of data to the host immediately upon receiving the first segment. The client will then allocate enough memory for the entire NetBIOS transaction, which we can then use to DMA the remainder of the data into as it arrives. In the case of a large (56k for example) NetBIOS session message, all but the first couple hundred bytes will be DMA'd to their final destination in memory.

Ex. 1031 (1997 Provisional FH) at .012.

PO's Cited Support Undermines Its Arguments Against APA and SMB



US 2002/0091844 A1

(19) **United States**

(12) **Patent Application Publication**

Craft et al.

(54) NETWORK INTERFACE DEVICE THAT FAST-PATH PROCESSES SOLICITED SESSION LAYER READ COMMANDS

(75) Inventors: Peter K. Craft, San Francisco, CA (US); Clive M. Philbrick, San Jose, CA (US); Laurence B. Boucher, Saratoga, CA (US)

Correspondence Address:
T. Lester Wallace
Patent Attorney
Suite 280
7041 Koll Center Parkway
Pleasanton, CA 94566 (US)

(73) Assignee: Alacritech, Inc.

(21) Appl. No.: 09/970,124

(22) Filed: Oct. 2, 2001

Related U.S. Application Data

(63) Continuation of application No. 08/967,511, filed on Apr. 27, 1998, now Pat. No. 6,226,680. Continuation of application No. 09/141,713, filed on Aug. 28, 1998, now Pat. No. 6,389,479. Continuation of application No. 09/384,792, filed on Aug. 27, 1999. Continuation of application No. 09/416,925, filed on Oct. 13, 1999. Continuation of application No. 09/439,603, filed on Nov. 12, 1999, now Pat. No. 6,247,060. Continuation of application No. 09/464,283, filed on Dec. 15, 1999. Continuation of application No. 09/514,425, filed on Feb. 28, 2000. Continuation of application No. 09/675,484, filed on Sep. 29, 2000. Continuation of application No. 09/691,403, filed on Oct. 17, 2000. Continuation of application No. 09/697,361, filed on Oct. 18, 2000. Continuation of application No. 09/748,976, filed on Dec. 26, 2000, now Pat. No.

(51) Int. Cl.

(52) U.S. Cl.

(57)

Alacritech, Ex. 2022.001

[0095] With INIC **606** operating on the client **602** when this reply arrives, the INIC **606** recognizes from the first frame received that this connection is receiving fast-path **620** processing (TCP/IP, NetBios, matching a CCB), and the SMB **616** may use this first frame to acquire buffer space for the message. The allocation of buffers can be provided by passing the first 192 bytes of the of the frame, including any NetBios/SMB headers, via the ATCP fast-path **620** directly to the client NetBios **614** to give NetBios/SMB the appropriate headers. NetBios/SMB will analyze these headers, realize by matching with a request ID that this is a reply to the original Read connection, and give the ATCP command driver a 64K list of buffers in a client file cache into which to place the data. At this stage only one frame has arrived,

Ex. 2022 (US 2002/0091844) ¶ 91
(cited in support of PO amendments at Paper 20, Appx. A).

POSA Would Be Motivated To Combine Thia, SMB And APA

- Thia and SMB teach using DMA to transfer data from a network interface to host memory
  - Thia and SMB teach using DMA engine to receive bulk data
  - APA teaches that Windows NT allocates and provided host destination address for that received data

See Ex. 1305 (Lin Decl. ISO Petitioner's Opp. To Mtn. to Amend) ¶ 22.

POSA Would Be Motivated To Combine Thia, SMB And APA

- POSA would have been motivated to combine Thia with popular Windows NT (described in APA) and SMB
  - APA's Windows NT was a widely used and very well-known operating system

    Ex. 1305 (Lin Decl. ISO Petitioner's Opp. To Mtn. to Amend) ¶ 22.
  - Thia and SMB teach reducing number of copies by directly placing data in host memory, which APA admits was desirable

    See id. ¶¶ 35-36.
  - Easily implemented features of popular software with predictable results

    See id. ¶ 33, A-29
  - APA's Windows NT is compatible with SMB

    Id. ¶ 33.

The Combination Renders Obvious "…to set up a fast-path connection"

- Thia sets up the fast-path connection while processing first packet received in response to a SMB read request, because this packet confirms connection is in data transfer phase
- Thia's DMA engine must be programmed with address on host memory for received data during fast-path processing
- It would be obvious to use the existing APA Windows NT feature to procure an address for DMA
  - Advantageous because amount of data to be received is identified in header of first response SMB packet

Ex. 1305 (Lin Decl. ISO Petitioner's Opp. To Mtn. to Amend) at A-20 – A-22;
Ex. 1031 (1997 Provisional FH) at .012.

The Combination Renders Obvious "…to set up a fast-path connection"

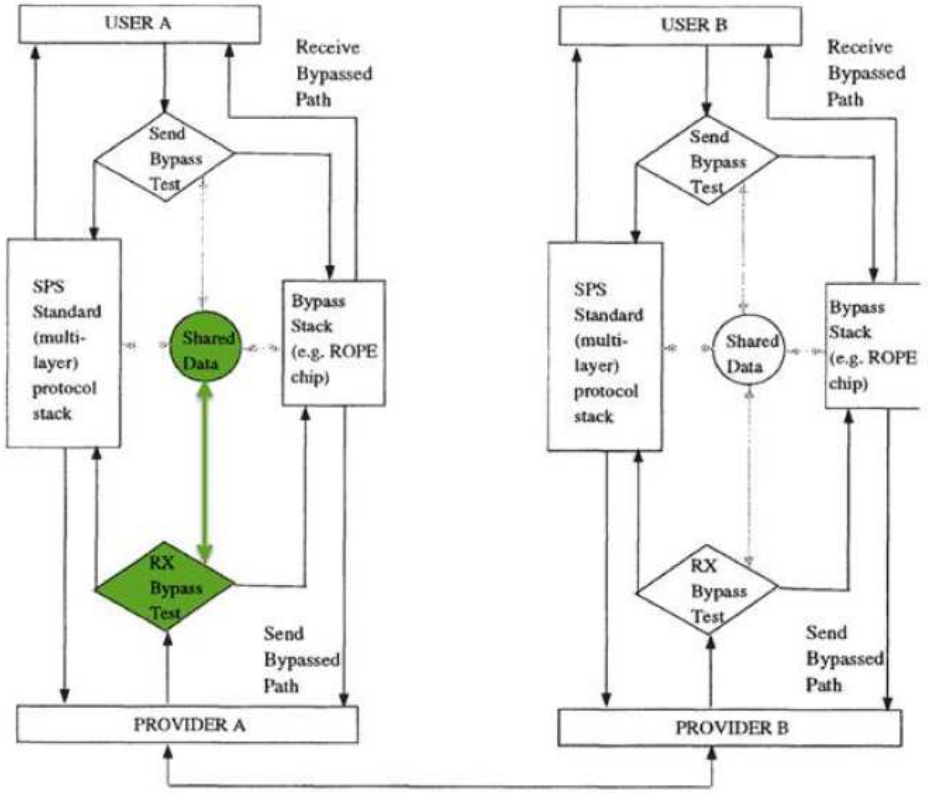


Figure 1 Bypass Architecture

- Receive bypass test shares data with the host
- It would have been obvious for receive bypass test to use a memory address supplied by the host to DMA received data

Ex. 1305 (Lin Decl. ISO Petitioner's Opp. To Mtn. to Amend) at A-27; Ex. 1015.003, .011 (Thia).

U.S. Patent No. 7,945,699
(699 Patent)

IPR2018-00401 (Cavium)
IPR2018-01352 (Intel)

*All citations herein are to the IPR2018-00401 case unless otherwise noted.

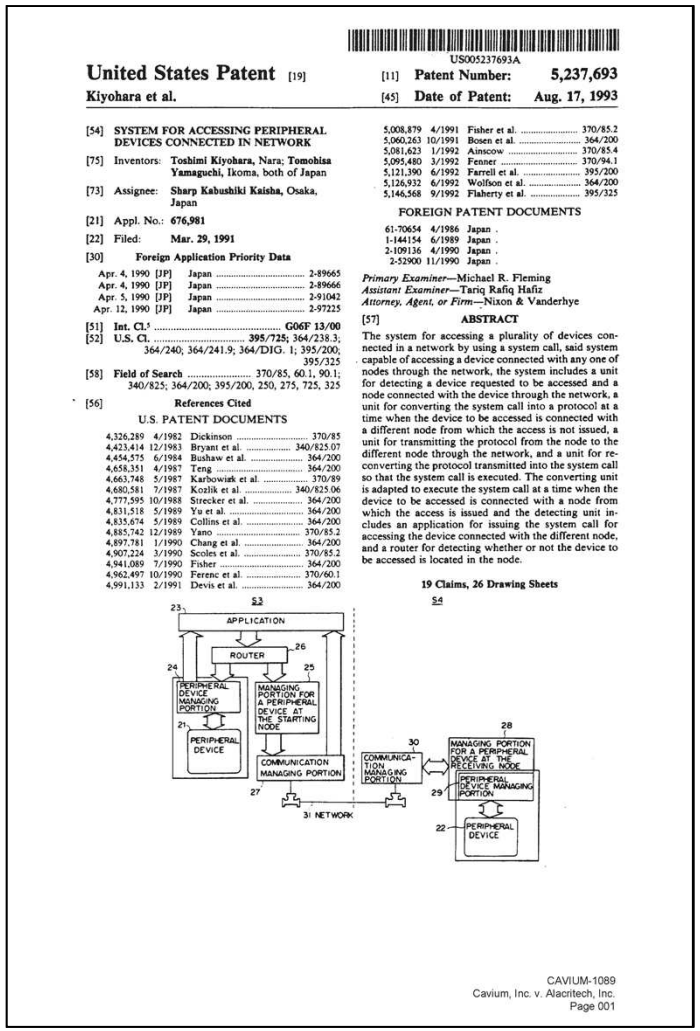
699 Patent: Instituted Ground

Kiyohara and **SMB**: claims **1**-3, 6, **7**, 10, 11, **13**, 16, and 17

Paper 8 (Institution Decision) at 12.

Tutorial on Prior Art References

- U.S. Patent No. 5,237,693 (“Kiyohara”) (Ex. 1089)



US005237693A

United States Patent [19]

Kiyohara et al.

[11] **Patent Number: 5,237,693**

[45] **Date of Patent: Aug. 17, 1993**

[54] **SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK**

[75] Inventors: **Toshimi Kiyohara**, Nara; **Tomohisa Yamaguchi**, Ikoma, both of Japan

[73] Assignee: **Sharp Kabushiki Kaisha**, Osaka, Japan

[21] Appl. No.: **676,981**

[22] Filed: **Mar. 29, 1991**

[30] **Foreign Application Priority Data**

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] **Int. Cl.**[5] **G06F 13/00**

[52] **U.S. Cl.** **395/725**; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325

[58] **Field of Search** 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,326,289 4/1982 Dickinson 370/85
4,423,414 12/1983 Bryant et al. 340/825.07
4,454,575 6/1984 Bushaw et al. 364/200
4,658,351 4/1987 Teng 364/200
4,663,748 5/1987 Karbowiak et al. 370/89
4,680,581 7/1987 Kozlik et al. 340/825.06
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200
5,008,879 4/1991 Fisher et al. 370/85.2
5,060,263 10/1991 Bosen et al. 364/200
5,081,623 1/1992 Ainscow 370/85.4
5,095,480 3/1992 Fenner 370/94.1
5,121,390 6/1992 Farrell et al. 395/200
5,126,932 6/1992 Wolfson et al. 364/200
5,146,568 9/1992 Flaherty et al. 395/325

FOREIGN PATENT DOCUMENTS

61-70654 4/1986 Japan .
1-144154 6/1989 Japan .
2-109136 4/1990 Japan .
2-52900 11/1990 Japan .

Primary Examiner—Michael R. Fleming
Assistant Examiner—Tariq Rafiq Hafiz
Attorney, Agent, or Firm—Nixon & Vanderhye

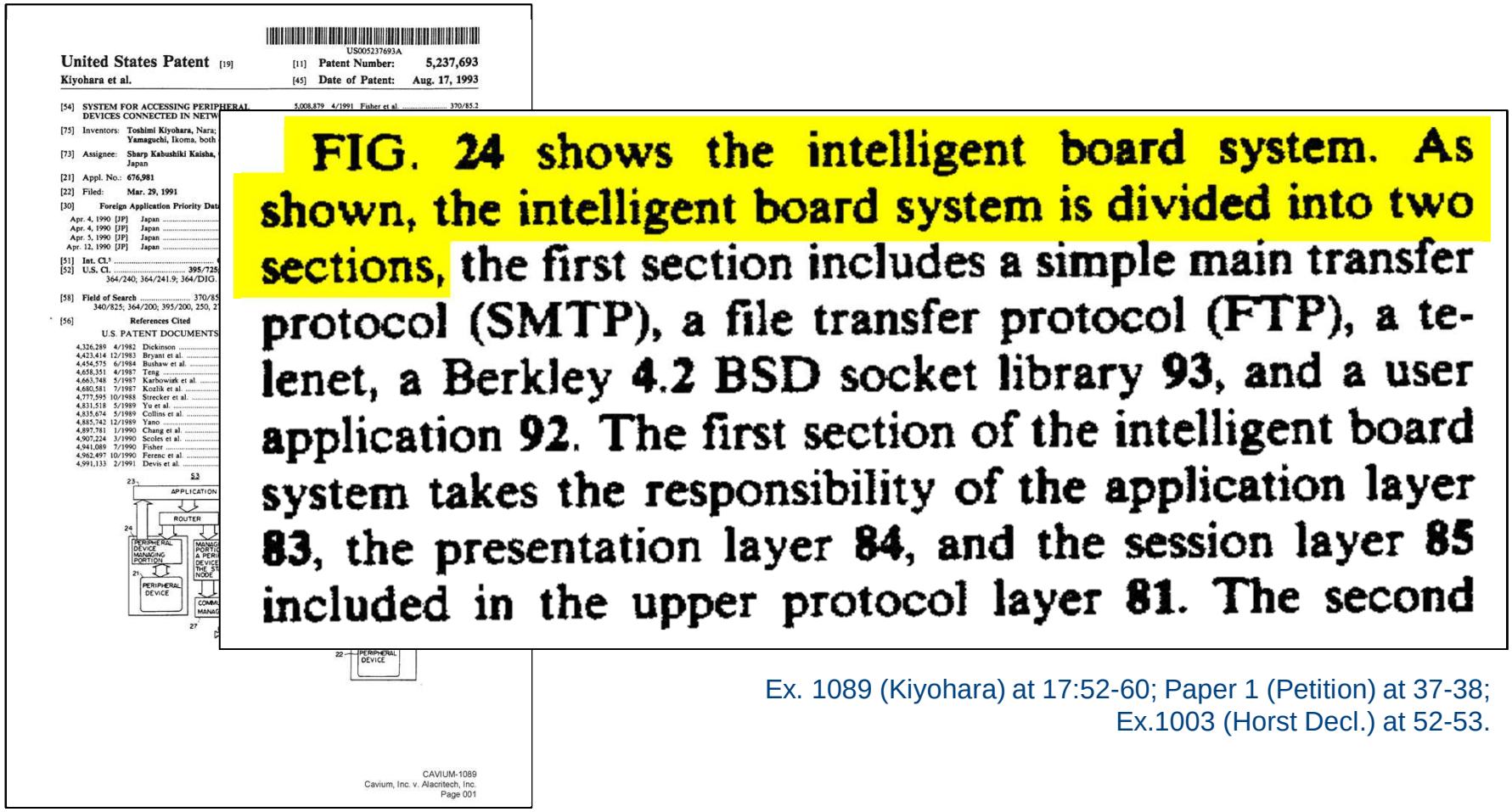
[57] **ABSTRACT**

The system for accessing a plurality of devices connected in a network by using a system call, said system capable of accessing a device connected with any one of nodes through the network, the system includes a unit for detecting a device requested to be accessed and a node connected with the device through the network, a unit for converting the system call into a protocol at a time when the device to be accessed is connected with a different node from which the access is not issued, a unit for transmitting the protocol from the node to the different node through the network, and a unit for re-converting the protocol transmitted into the system call so that the system call is executed. The converting unit is adapted to execute the system call at a time when the device to be accessed is connected with a node from which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

19 Claims, 26 Drawing Sheets

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Kiyohara teaches an intelligent board system with two sections



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number:** 5,237,693
[45] **Date of Patent:** Aug. 17, 1993

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

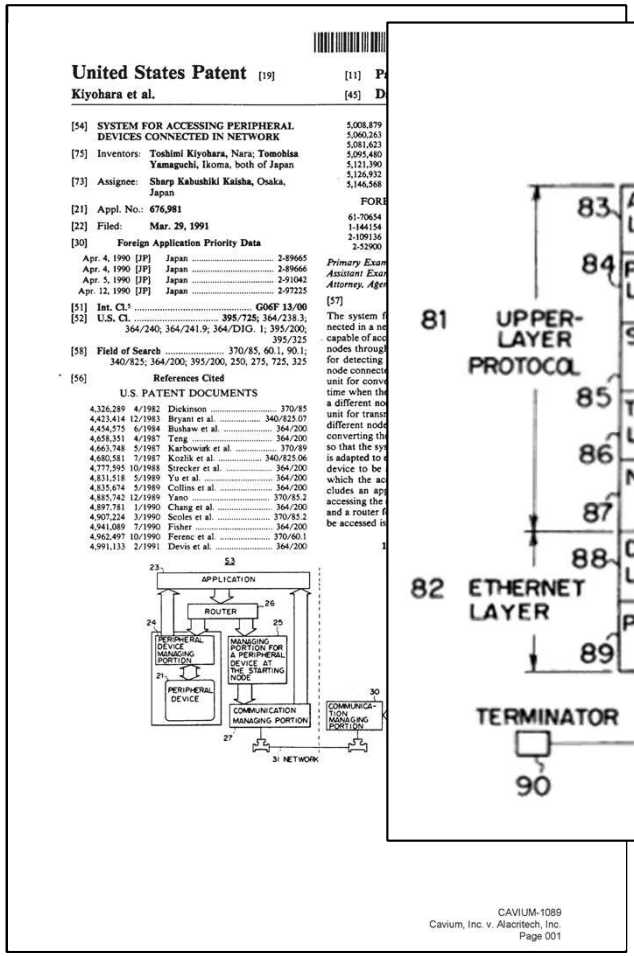
4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al.
5,008,879 4/1991 Fisher et al. 370/85.2

FIG. 24 shows the intelligent board system. As shown, the intelligent board system is divided into two sections, the first section includes a simple main transfer protocol (SMTP), a file transfer protocol (FTP), a telenet, a Berkley 4.2 BSD socket library 93, and a user application 92. The first section of the intelligent board system takes the responsibility of the application layer 83, the presentation layer 84, and the session layer 85 included in the upper protocol layer 81. The second

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Ex. 1089 (Kiyohara) at 17:52-60; Paper 1 (Petition) at 37-38;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara teaches an intelligent board system with two sections



United States Patent [19]
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] Int. Cl.[5] G06F 13/00
[52] U.S. Cl. 395/725; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] Field of Search 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] References Cited

U.S. PATENT DOCUMENTS

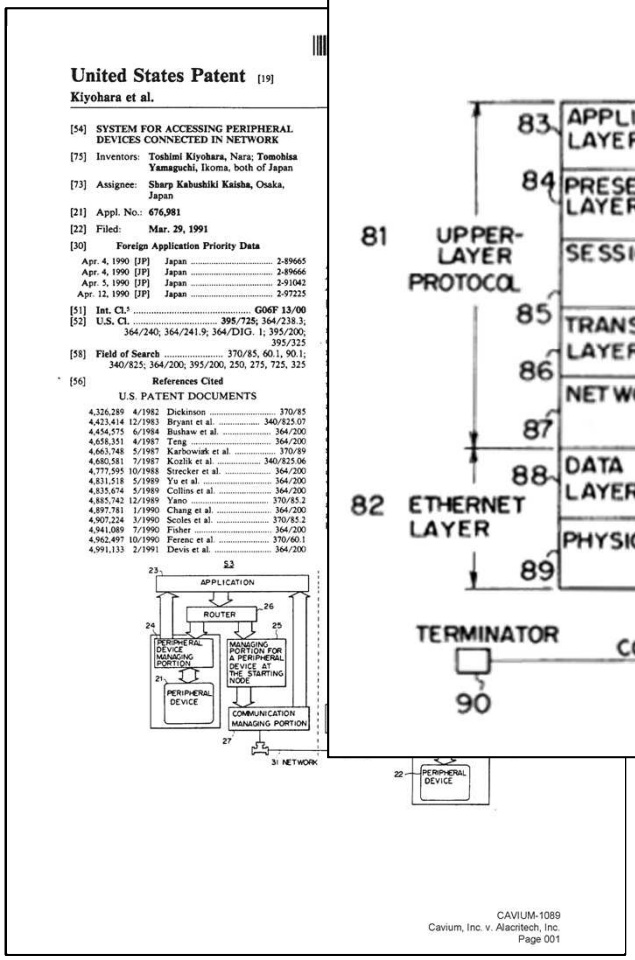
4,326,289 4/1982 Dickinson 370/85
4,423,414 12/1983 Bryant et al. 340/825.07
4,454,575 6/1984 Bushaw et al. 364/200
4,658,351 4/1987 Teng 364/200
4,663,748 5/1987 Karbowiak et al. 370/89
4,680,581 7/1987 Kozlik et al. 340/825.06
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Fig. 24

Ex. 1089 (Kiyohara) at Fig. 24; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

A host bus connects the two sections



United States Patent [19]
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] Int. Cl.[5] G06F 13/00
[52] U.S. Cl. 395/725; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] Field of Search 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] References Cited

U.S. PATENT DOCUMENTS

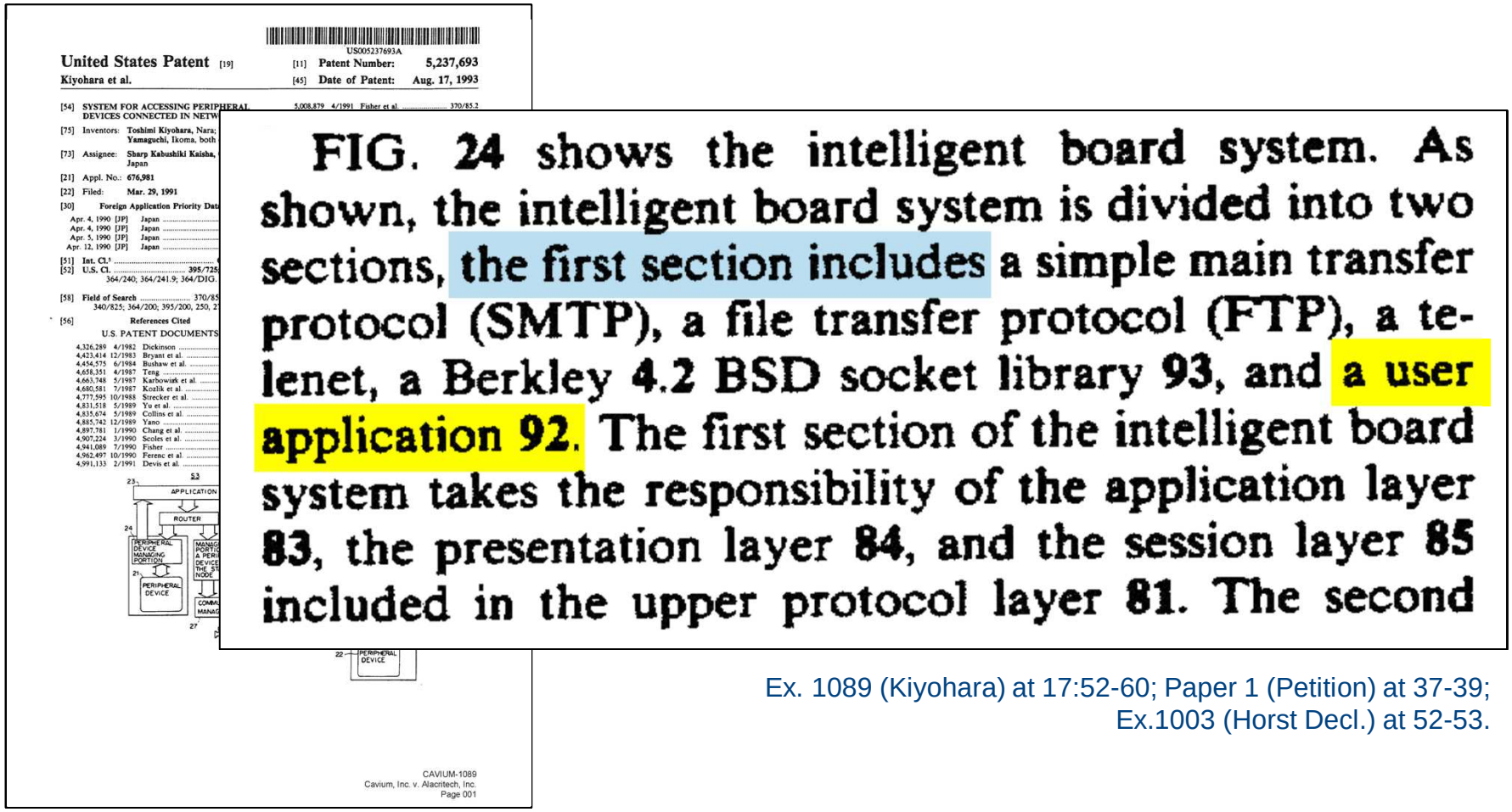
4,326,289	4/1982	Dickinson	370/85
4,423,414	12/1983	Bryant et al.	340/825.07
4,454,575	6/1984	Bushaw et al.	364/200
4,658,351	4/1987	Teng	364/200
4,663,748	5/1987	Karbowiak et al.	370/89
4,680,581	7/1987	Kozlik et al.	340/825.06
4,777,595	10/1988	Strecker et al.	364/200
4,831,518	5/1989	Yu et al.	364/200
4,835,674	5/1989	Collins et al.	364/200
4,885,742	12/1989	Yano	370/85.2
4,897,781	1/1990	Chang et al.	364/200
4,907,224	3/1990	Scoles et al.	370/85.2
4,941,089	7/1990	Fisher	364/200
4,962,497	10/1990	Ferenc et al.	370/60.1
4,991,133	2/1991	Devis et al.	364/200

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Fig. 24

Ex. 1089 (Kiyohara) at Fig. 24; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's first section includes a user application



United States Patent [19]
Kiyohara et al.

US005237693A
[11] Patent Number: 5,237,693
[45] Date of Patent: Aug. 17, 1993

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat
Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725
364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85
340/825; 364/200; 395/200, 250, 2

[56] References Cited
U.S. PATENT DOCUMENTS

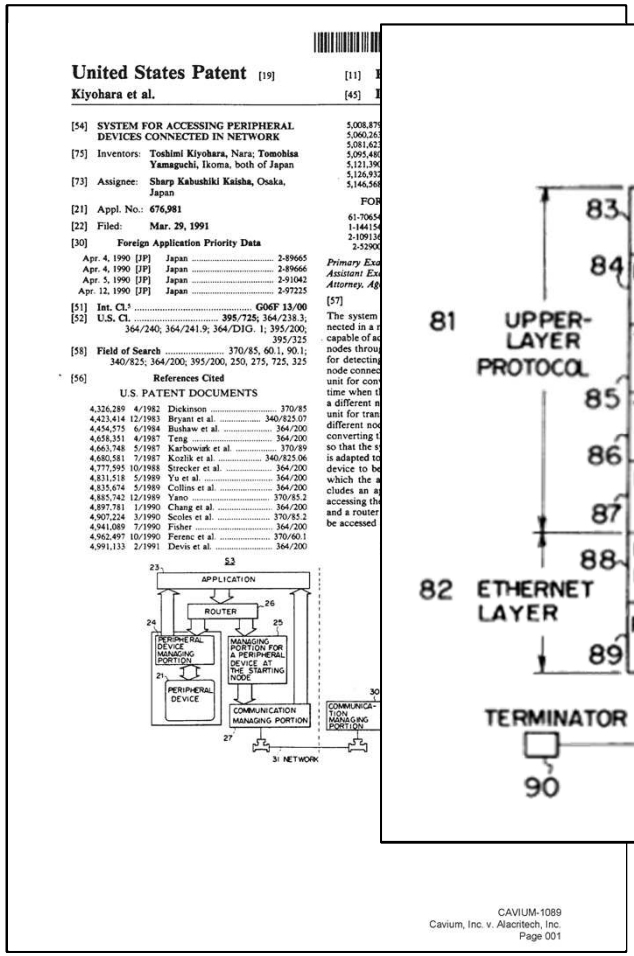
4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al.
5,008,879 4/1991 Fisher et al. 370/85.2

FIG. 24 shows the intelligent board system. As shown, the intelligent board system is divided into two sections, the first section includes a simple main transfer protocol (SMTP), a file transfer protocol (FTP), a telenet, a Berkley 4.2 BSD socket library 93, and a user application 92. The first section of the intelligent board system takes the responsibility of the application layer 83, the presentation layer 84, and the session layer 85 included in the upper protocol layer 81. The second

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Ex. 1089 (Kiyohara) at 17:52-60; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's first section includes a user application



United States Patent [19]
Kiyohara et al.

[11] [45]

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] Int. Cl.[5] G06F 13/00
[52] U.S. Cl. 395/725; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] Field of Search 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] References Cited

U.S. PATENT DOCUMENTS

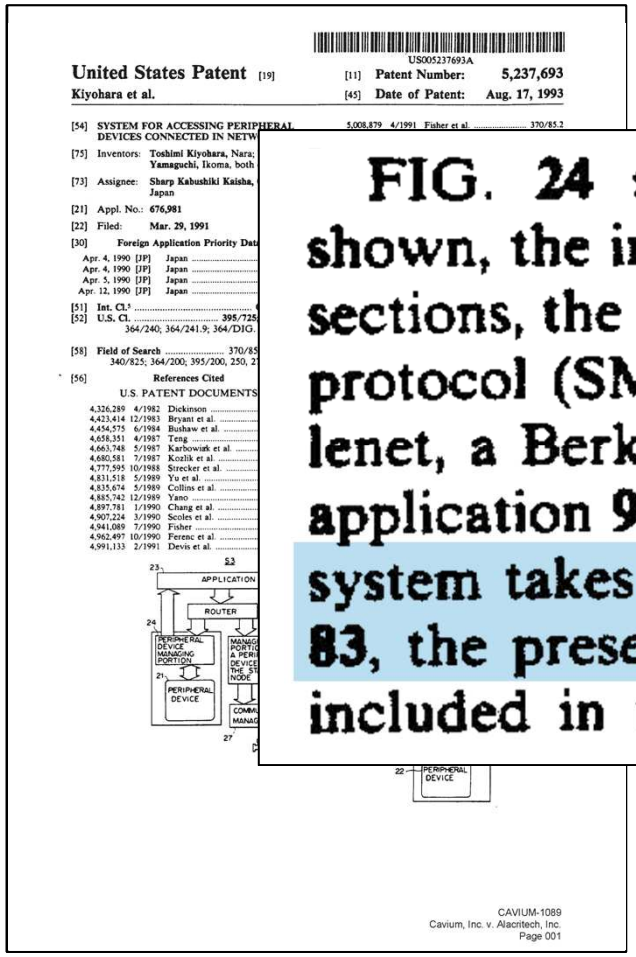
4,326,289 4/1982 Dickinson 370/85
4,423,414 12/1983 Bryant et al. 340/825.07
4,454,575 6/1984 Bushaw et al. 364/200
4,658,351 4/1987 Teng 364/200
4,663,748 5/1987 Karbowiak et al. 370/89
4,680,581 7/1987 Kozlik et al. 340/825.06
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Fig. 24

Ex. 1089 (Kiyohara) at Fig. 24; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's first section processes the application, presentation & session layers of the protocol stack



United States Patent [19]
Kiyohara et al.

US005237693A
[11] Patent Number: 5,237,693
[45] Date of Patent: Aug. 17, 1993

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85; 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al.

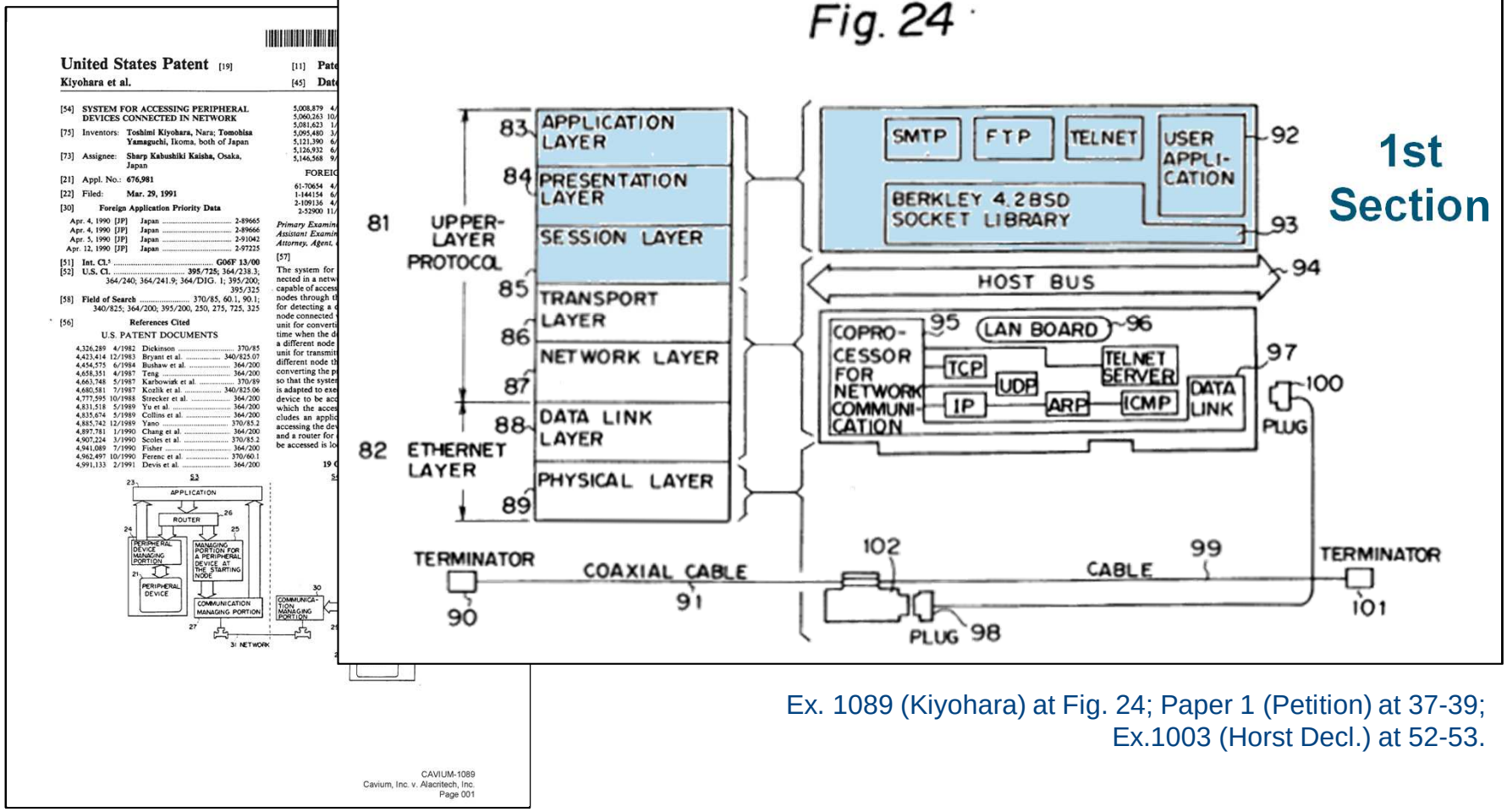
5,008,879 4/1991 Fisher et al. 370/85.2

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

FIG. 24 shows the intelligent board system. As shown, the intelligent board system is divided into two sections, the first section includes a simple main transfer protocol (SMTP), a file transfer protocol (FTP), a telenet, a Berkley 4.2 BSD socket library 93, and a user application 92. The first section of the intelligent board system takes the responsibility of the application layer 83, the presentation layer 84, and the session layer 85 included in the upper protocol layer 81. The second

Ex. 1089 (Kiyohara) at 17:52-60; Paper 1 (Petition) at 37-38;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's first section processes the application, presentation & session layers of the protocol stack



United States Patent [19]
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] Int. Cl.[5] G06F 13/00
[52] U.S. Cl. 395/725; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] Field of Search 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] References Cited

U.S. PATENT DOCUMENTS

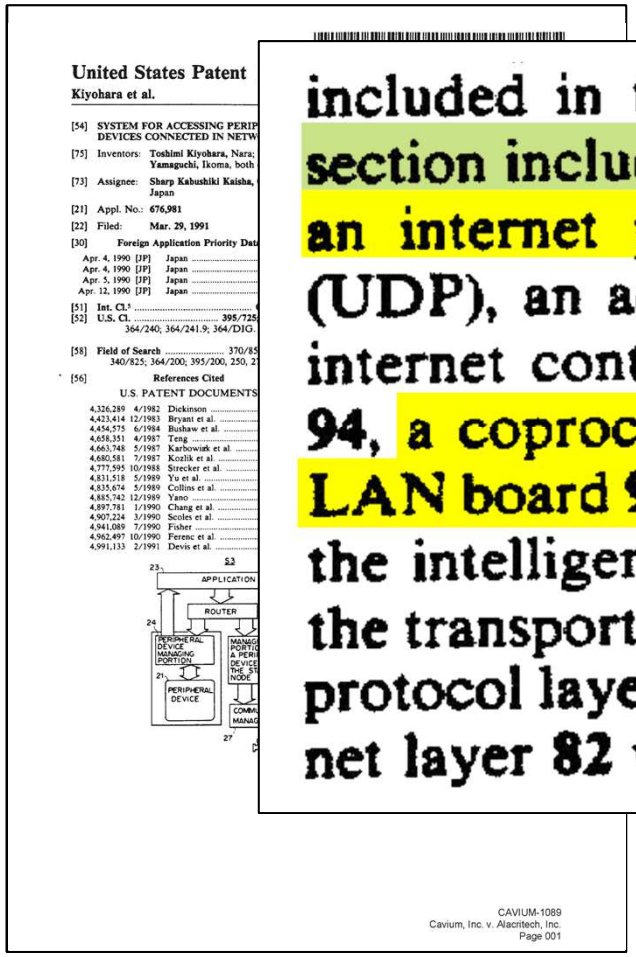
4,326,289 4/1982 Dickinson 370/85
4,423,414 12/1983 Bryant et al. 340/825.07
4,454,575 6/1984 Bushaw et al. 364/200
4,658,351 4/1987 Teng 364/200
4,663,748 5/1987 Karbowiak et al. 370/89
4,680,581 7/1987 Kozlik et al. 340/825.06
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Fig. 24

Ex. 1089 (Kiyohara) at Fig. 24; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's second section includes TCP and IP protocols, a network coprocessor, & a LAN card


United States Patent
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIP
DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725
364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85
340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

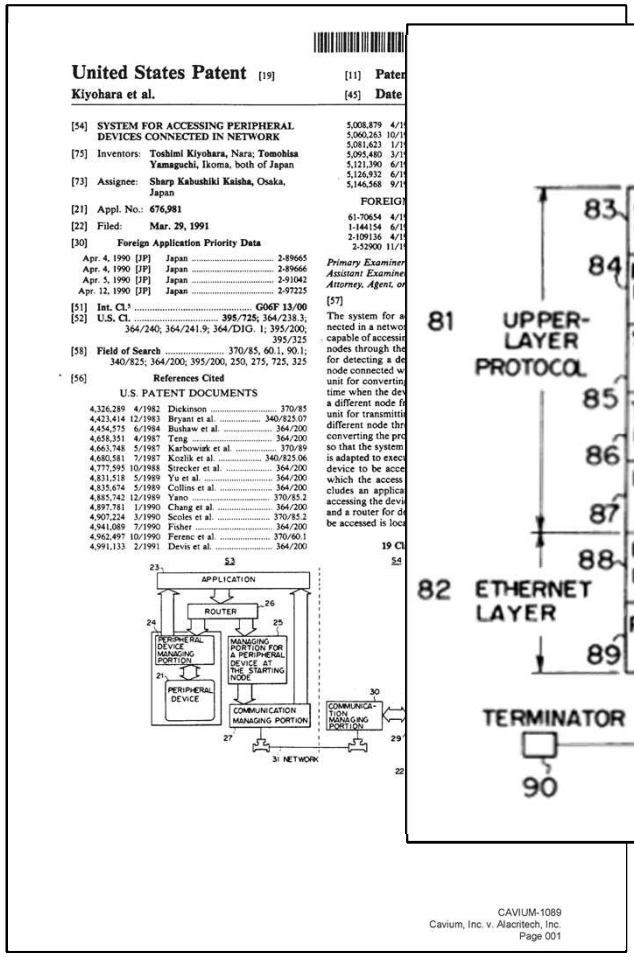
4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al.

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

included in the upper protocol layer **81**. The second section includes a transmission control protocol (TCP), an internet protocol (IP), a user datagram protocol (UDP), an address resolution protocol (ARP), and an internet control message protocol (ICMP), a host bus **94**, a coprocessor for a network communication **95**, a LAN board **96**, and a data link **97**. The second section of the intelligent board system takes the responsibility of the transport layer **86**, the network layer **87** in the upper protocol layer **81**, and the data link layer **88** in the ether-net layer **82** which includes a physical layer **89**.

Ex. 1089 (Kiyohara) at 17:60-18:2; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's second section includes TCP and IP protocols, a network coprocessor, & a LAN card



United States Patent [19]
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] Int. Cl.[5] G06F 13/00
[52] U.S. Cl. 395/725; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] Field of Search 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] References Cited

U.S. PATENT DOCUMENTS

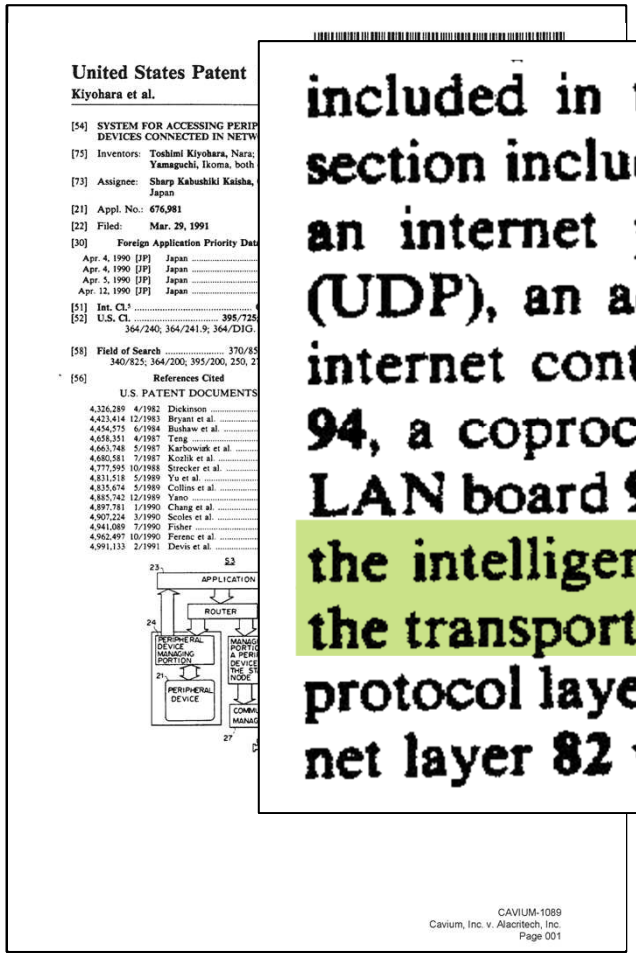
4,326,289 4/1982 Dickinson 370/85
4,423,414 12/1983 Bryant et al. 340/825.07
4,454,575 6/1984 Bushaw et al. 364/200
4,658,351 4/1987 Teng 364/200
4,663,748 5/1987 Karbowiak et al. 370/89
4,680,581 7/1987 Kozlik et al. 340/825.06
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Fig. 24

Ex. 1089 (Kiyohara) at Fig. 24; Paper 1 (Petition) at 37-39; Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's second section is responsible for the transport & network layers of the protocol stack


United States Patent
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIP
DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725
364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85
340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

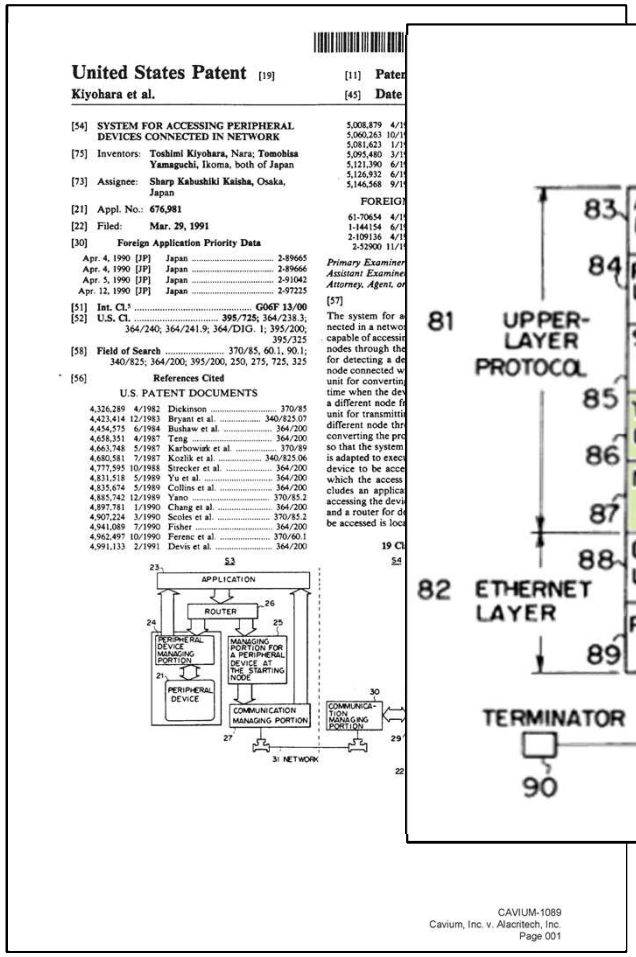
4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al.

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

included in the upper protocol layer **81**. The second section includes a transmission control protocol (TCP), an internet protocol (IP), a user datagram protocol (UDP), an address resolution protocol (ARP), and an internet control message protocol (ICMP), a host bus **94**, a coprocessor for a network communication **95**, a LAN board **96**, and a data link **97**. The second section of the intelligent board system takes the responsibility of the transport layer **86**, the network layer **87** in the upper protocol layer **81**, and the data link layer **88** in the ether-net layer **82** which includes a physical layer **89**.

Ex. 1089 (Kiyohara) at 17:60-18:2; Paper 1 (Petition) at 37-39;
Ex.1003 (Horst Decl.) at 52-53.

Kiyohara's second section is responsible for the transport & network layers of the protocol stack



United States Patent [19]
Kiyohara et al.

[11] Pater
[45] Date

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] Int. Cl.[5] G06F 13/00
[52] U.S. Cl. 395/725; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] Field of Search 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] References Cited

U.S. PATENT DOCUMENTS

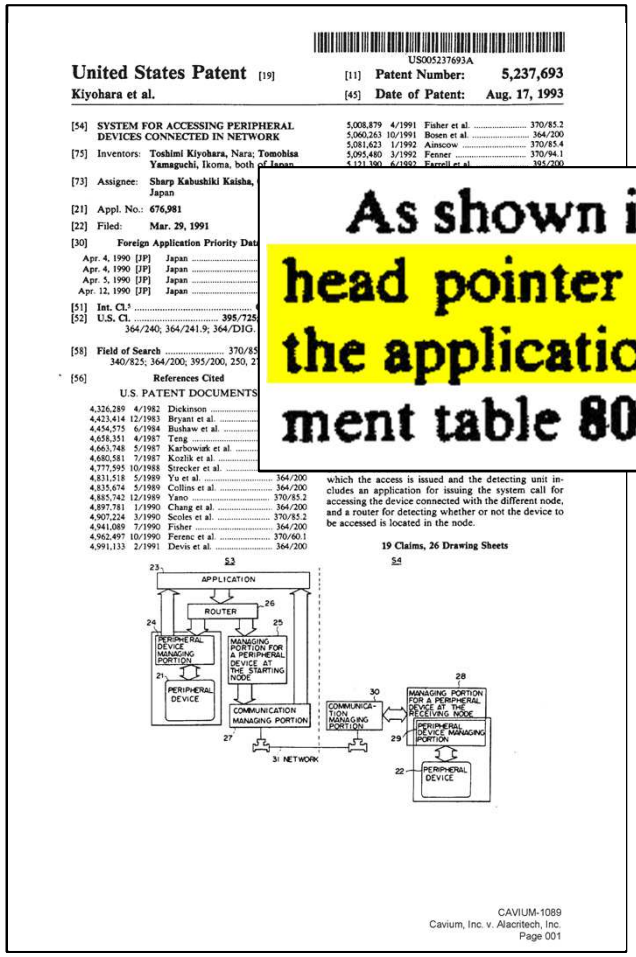
4,326,289 4/1982 Dickinson 370/85
4,423,414 12/1983 Bryant et al. 340/825.07
4,454,575 6/1984 Bushaw et al. 364/200
4,658,351 4/1987 Teng 364/200
4,663,748 5/1987 Karbowiak et al. 370/89
4,680,581 7/1987 Kozlik et al. 340/825.06
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

Fig. 24

Ex. 1089 (Kiyohara) at Fig. 24; Paper 1 (Petition) at 37-39; Ex.1003 (Horst Decl.) at 52-53.

Kiyohara obtains a pointer to the data storage area from the application



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.[5]
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85; 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

5,008,879 4/1991 Fisher et al. 370/85.2
5,060,263 10/1991 Bosen et al. 364/200
5,081,623 1/1992 Ainscow 370/85.4
5,095,480 3/1992 Fenner 370/94.1
5,121,390 6/1992 Farrell et al. 395/200

which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

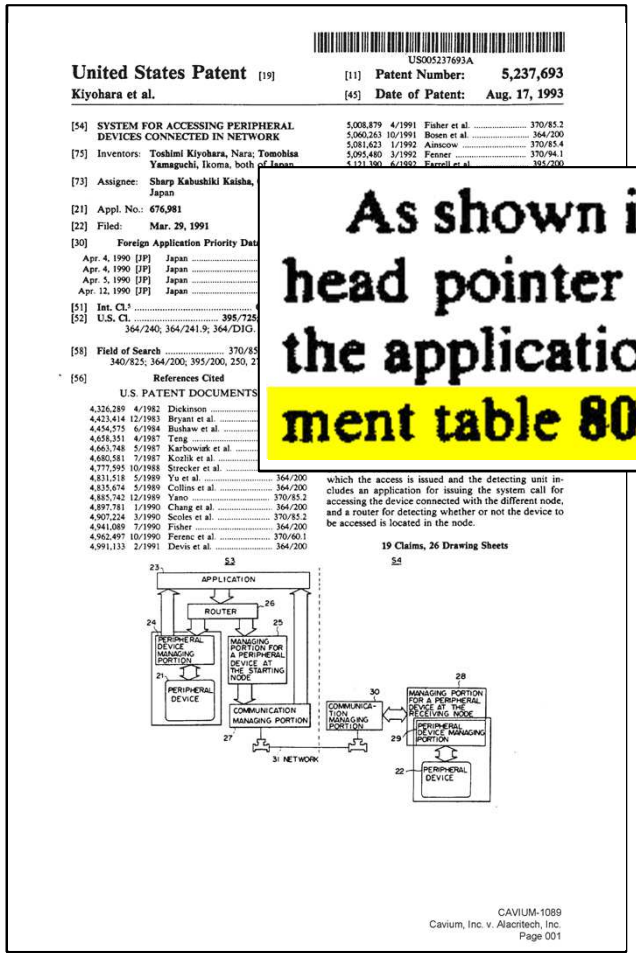
19 Claims, 26 Drawing Sheets

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

As shown in FIG. 22A, like the data transmission, the head pointer of the data storage area is obtained from the application (step SX1) and the data pointer management table 80 is created (step SX2). Then, when receiv-

Ex. 1089 (Kiyohara) at 16:60-63;
Paper 1 (Petition) at 39-42;
Ex.1003 (Horst Decl.) at 58.

Kiyohara then creates the data pointer management table



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomohisa Yamaguchi, Ikoma, both of Japan

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.[5]
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85; 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

5,008,879 4/1991 Fisher et al. 370/85.2
5,060,263 10/1991 Bosen et al. 364/200
5,081,623 1/1992 Ainscow 370/85.4
5,095,480 3/1992 Fenner 370/94.1
5,121,390 6/1992 Farrell et al. 395/200

which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

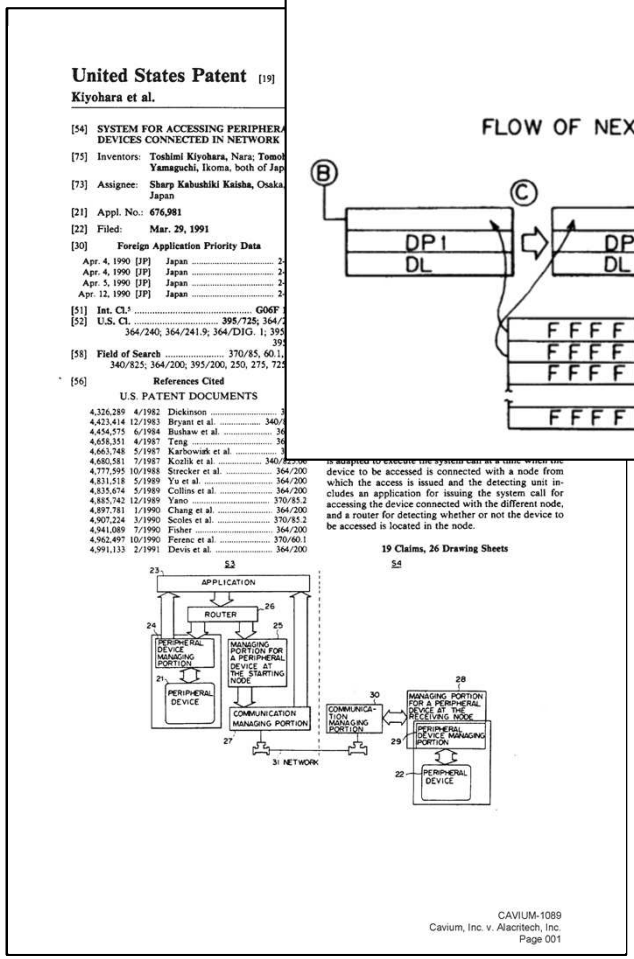
19 Claims, 26 Drawing Sheets

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

As shown in FIG. 22A, like the data transmission, the head pointer of the data storage area is obtained from the application (step SX1) and the data pointer management table 80 is created (step SX2). Then, when receiv-

Ex. 1089 (Kiyohara) at Fig. 22A, 16:60-63;
Paper 1 (Petition) at 39-42;
Ex.1003 (Horst Decl.) at 58.

Kiyohara then creates the data pointer management table that manages pointers to data for each packet



United States Patent [19]
Kiyohara et al.

[54] SYSTEM FOR ACCESSING PERIPHERA DEVICES CONNECTED IN NETWORK

[75] Inventors: Toshimi Kiyohara, Nara; Tomok Yamaguchi, Ikoma, both of Jap

[73] Assignee: Sharp Kabushiki Kaisha, Osaka, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan 2-
Apr. 4, 1990 [JP] Japan 2-
Apr. 5, 1990 [JP] Japan 2-
Apr. 12, 1990 [JP] Japan 2-

[51] Int. Cl.[5] G06F
[52] U.S. Cl. 395/725; 364/2 364/240; 364/241.9; 364/DIG. 1; 395

[58] Field of Search 370/85, 60.1, 340/825; 364/200; 395/200, 250, 275, 72

[56] References Cited

U.S. PATENT DOCUMENTS

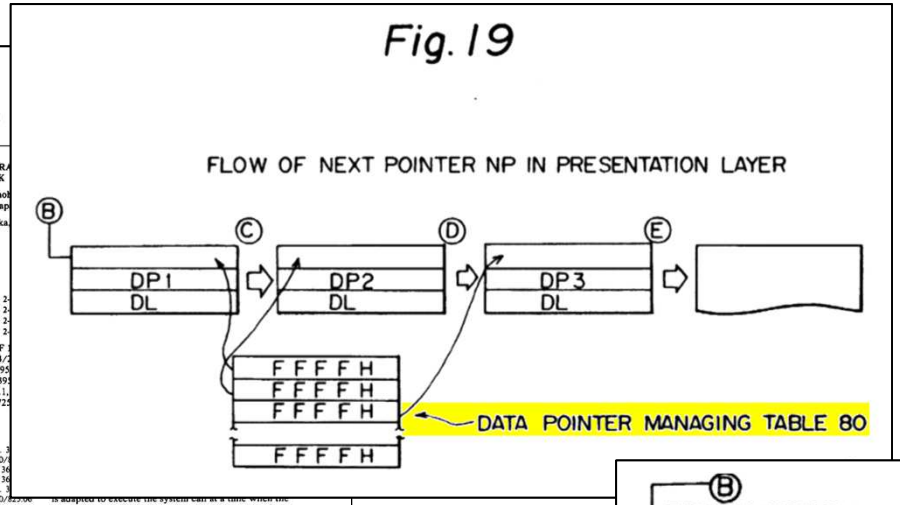
4,326,289 4/1982 Dickinson 3
4,423,414 12/1983 Bryant et al. 340/
4,454,575 6/1984 Bushaw et al. 36
4,658,351 4/1987 Teng 36
4,663,748 5/1987 Karbowiak et al. 7
4,680,581 7/1987 Kozlik et al. 340/
4,777,595 10/1988 Strecker et al. 364/200
4,831,518 5/1989 Yu et al. 364/200
4,835,674 5/1989 Collins et al. 364/200
4,885,742 12/1989 Yano 370/85.2
4,897,781 1/1990 Chang et al. 364/200
4,907,224 3/1990 Scoles et al. 370/85.2
4,941,089 7/1990 Fisher 364/200
4,962,497 10/1990 Ferenc et al. 370/60.1
4,991,133 2/1991 Devis et al. 364/200

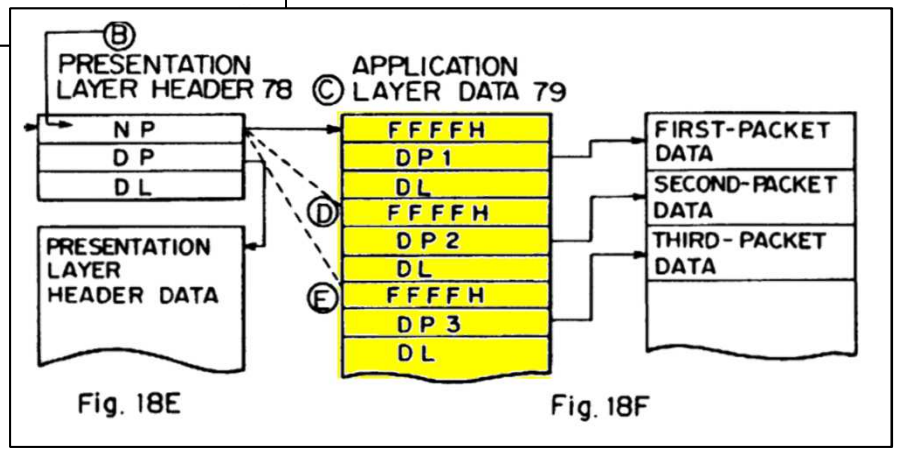
device to be accessed is connected with a node from which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

19 Claims, 26 Drawing Sheets

53 54

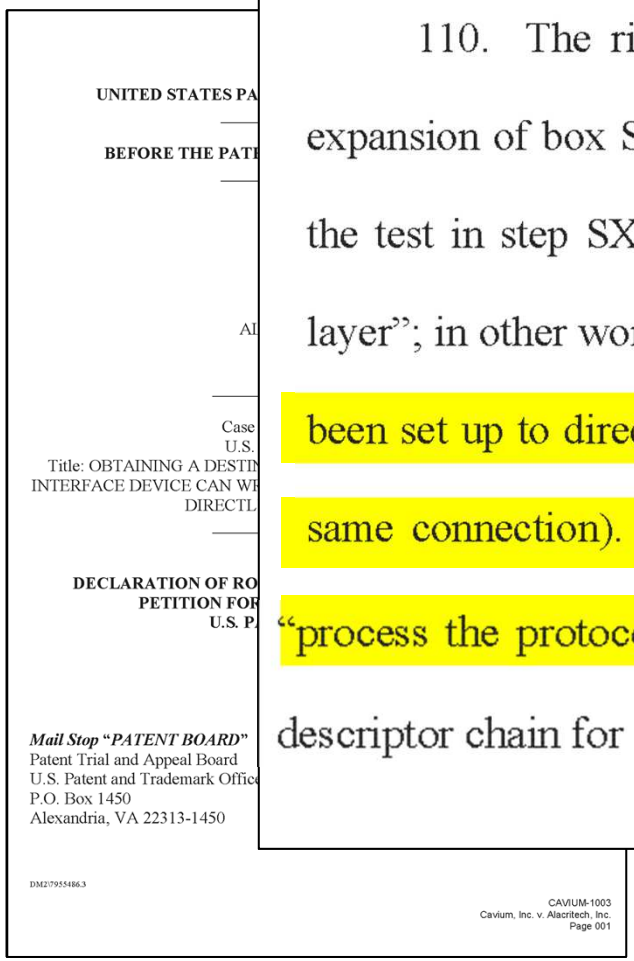
CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001





Ex. 1089 (Kiyohara) at Fig. 18E, 18F, 19;
Paper 1 (Petition) at 41-44, FN 8; Ex.1003 (Horst Decl.) at 56-58.

Kiyohara processes each upper layer if the descriptor chain for the received packet has not been set up


UNITED STATES PA

BEFORE THE PATE

AI

Case
U.S.
Title: OBTAINING A DESTIN
INTERFACE DEVICE CAN WF
DIRECTL

DECLARATION OF RO
PETITION FOR
U.S. P.

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

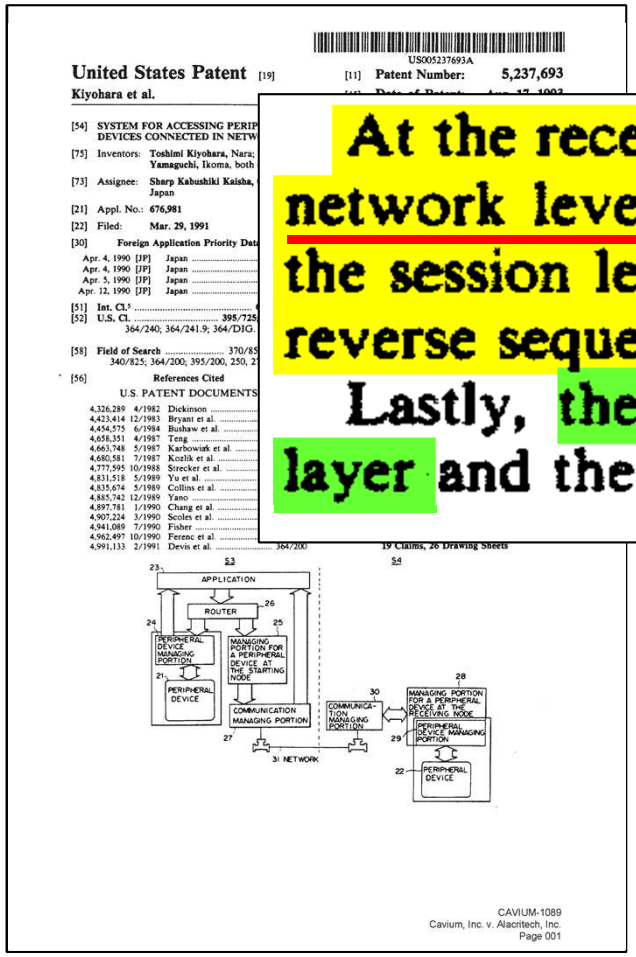
DM2\7955486.3

CAVIUM-1003
Cavium, Inc. v. Alacritech, Inc.
Page 001

110. The right side of Fig. 22 includes steps SC41-SX45 which are an expansion of box SX4 on the left side of the diagram. When a packet is received, the test in step SX42 determines whether the "data has the same header at each layer"; in other words, the test determines whether the descriptor chain has already been set up to direct the received data to the data storage area in memory (for the same connection). In the case where this test fails, step SX44 is performed to "process the protocol at higher layers." Such processing includes setting up the descriptor chain for subsequent transfers.

Ex. 1003 (Horst Decl) at 58-59;
Paper 1 (Petition) at 42-44.

Kiyohara removes protocol headers in sequence leaving just the data at the application layer



United States Patent [19]
Kiyohara et al.

[11] Patent Number: 5,237,693

[54] SYSTEM FOR ACCESSING PERIP... DEVICES CONNECTED IN NETW...

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Data

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85; 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al. 364/200

19 Claims, 26 Drawing Sheets

53 54

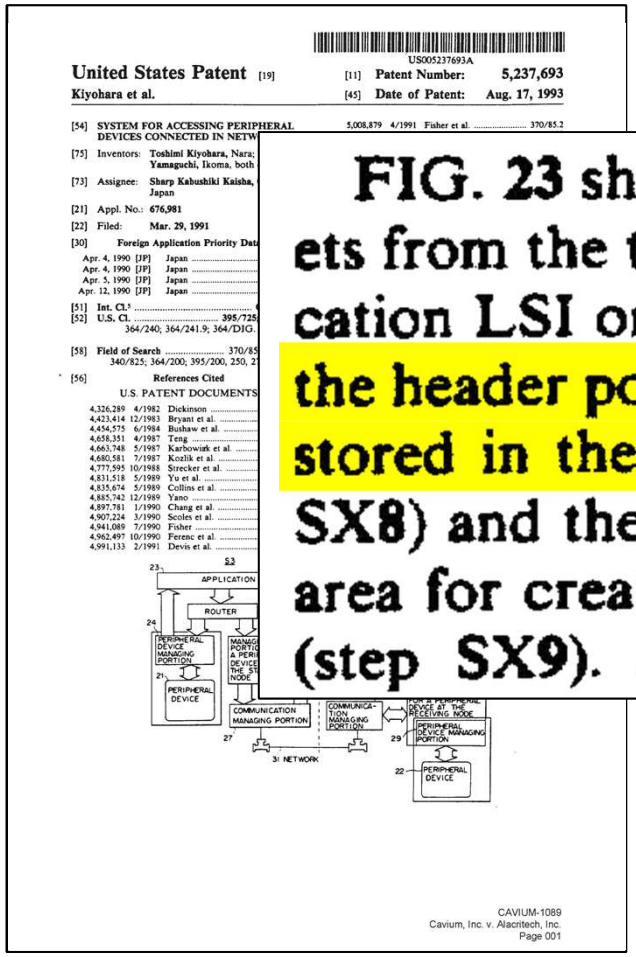
CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

At the receiving node, the data link level header, the network level header, the transport level header, and the session level header are removed from the data in reverse sequence.

Lastly, the data are represented at the application layer and the presentation layer.

Ex. 1089 (Kiyohara) at 14:51-56; Paper 1 (Petition) at 36-37;
Ex.1003 (Horst Decl.) at 54-58.

Kiyohara cuts out headers from packets and places them in a header storage area



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

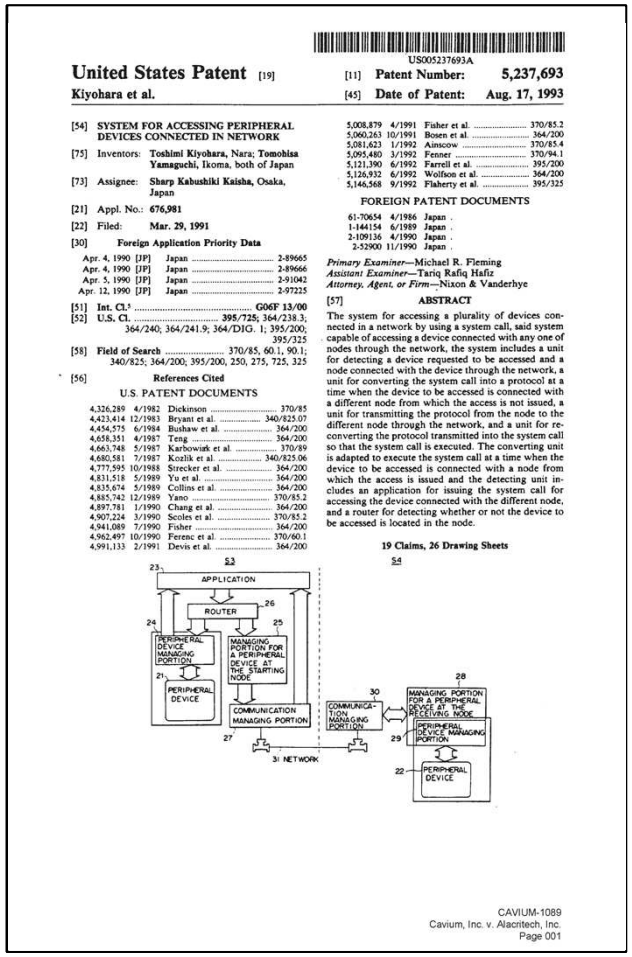
4,326,289	4/1982	Dickinson	
4,423,414	12/1983	Bryant et al.	
4,454,575	6/1984	Bushaw et al.	
4,658,351	4/1987	Teng	
4,663,748	5/1987	Karbowiak et al.	
4,680,581	7/1987	Kozlik et al.	
4,777,595	10/1988	Strecker et al.	
4,831,518	5/1989	Yu et al.	
4,835,674	5/1989	Collins et al.	
4,885,742	12/1989	Yano	
4,897,781	1/1990	Chang et al.	
4,907,224	3/1990	Scoles et al.	
4,941,089	7/1990	Fisher	
4,962,497	10/1990	Ferenc et al.	
4,991,133	2/1991	Devis et al.	
5,008,879	4/1991	Fisher et al.	370/85.2

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

FIG. 23 shows the procedure for receiving the packets from the transmission medium. When the communication LSI or hardware issues a receiving interruption, the header portion is cut out from the packet data and is stored in the header storage area for each layer (step SX8) and the data portion is stored in the data storage area for creating the management descriptor table DT (step SX9). After that, whether the communication

Ex. 1089 (Kiyohara) at 17:18-25; Paper 1 (Petition) at 44, 61;
Ex.1003 (Horst Decl.) at A-13.

Kiyohara creates the descriptor table and the data pointer management table for new connections



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] **SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK**

[75] Inventors: **Toshimi Kiyohara**, Nara; **Tomohisa Yamaguchi**, Ikoma, both of Japan

[73] Assignee: **Sharp Kabushiki Kaisha**, Osaka, Japan

[21] Appl. No.: **676,981**

[22] Filed: **Mar. 29, 1991**

[30] **Foreign Application Priority Data**

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] **Int. Cl.**[5] **G06F 13/00**
[52] **U.S. Cl.** **395/725**; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] **Field of Search** 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,326,289	4/1982	Dickinson	370/85
4,423,414	12/1983	Bryant et al.	340/825.07
4,454,575	6/1984	Bushaw et al.	364/200
4,658,351	4/1987	Teng	364/200
4,663,748	5/1987	Karbowiak et al.	370/89
4,680,581	7/1987	Kozlik et al.	340/825.06
4,777,595	10/1988	Strecker et al.	364/200
4,831,518	5/1989	Yu et al.	364/200
4,835,674	5/1989	Collins et al.	364/200
4,885,742	12/1989	Yano	370/85.2
4,897,781	1/1990	Chang et al.	364/200
4,907,224	3/1990	Scoles et al.	370/85.2
4,941,089	7/1990	Fisher	364/200
4,962,497	10/1990	Ferenc et al.	370/60.1
4,991,133	2/1991	Devis et al.	364/200
5,008,879	4/1991	Fisher et al.	370/85.2
5,060,263	10/1991	Bosen et al.	364/200
5,081,623	1/1992	Ainscow	370/85.4
5,095,480	3/1992	Fenner	370/94.1
5,121,390	6/1992	Farrell et al.	395/200
5,126,932	6/1992	Wolfson et al.	364/200
5,146,568	9/1992	Flaherty et al.	395/325

FOREIGN PATENT DOCUMENTS

61-70654	4/1986	Japan .
1-144154	6/1989	Japan .
2-109136	4/1990	Japan .
2-52900	11/1990	Japan .

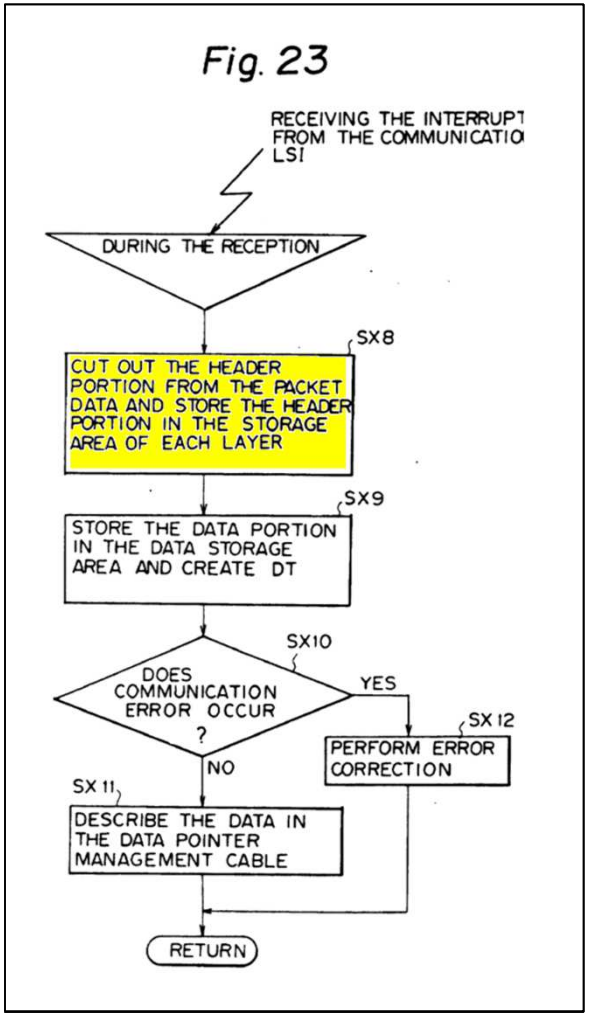
Primary Examiner—Michael R. Fleming
Assistant Examiner—Tariq Rafiq Hafiz
Attorney, Agent, or Firm—Nixon & Vanderhye

[57] **ABSTRACT**

The system for accessing a plurality of devices connected in a network by using a system call, said system capable of accessing a device connected with any one of nodes through the network, the system includes a unit for detecting a device requested to be accessed and a node connected with the device through the network, a unit for converting the system call into a protocol at a time when the device to be accessed is connected with a different node from which the access is not issued, a unit for transmitting the protocol from the node to the different node through the network, and a unit for re-converting the protocol transmitted into the system call so that the system call is executed. The converting unit is adapted to execute the system call at a time when the device to be accessed is connected with a node from which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

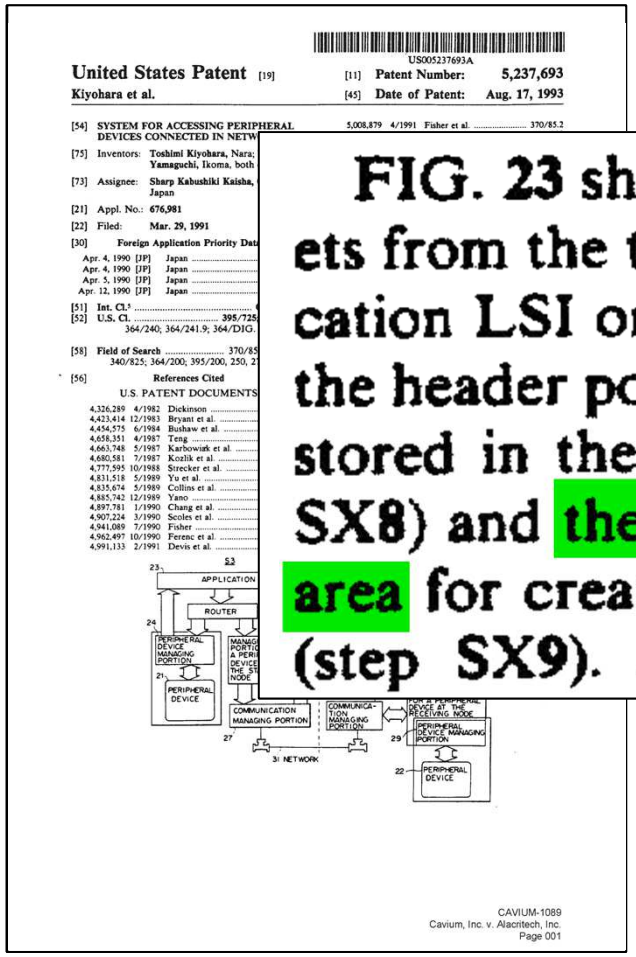
19 Claims, 26 Drawing Sheets

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001



Ex. 1089 (Kiyohara) at Fig. 23, 17:18-25;
Paper 1 (Petition) at 39-42; Ex.1003 (Horst Decl.) at A-13.

Kiyohara places the data portion in a data storage area



United States Patent [19]
Kiyohara et al.

[11] Patent Number: 5,237,693
[45] Date of Patent: Aug. 17, 1993

[54] SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETW

[75] Inventors: Toshimi Kiyohara, Nara; Yamaguchi, Ikoma, both

[73] Assignee: Sharp Kabushiki Kaisha, Japan

[21] Appl. No.: 676,981

[22] Filed: Mar. 29, 1991

[30] Foreign Application Priority Dat

Apr. 4, 1990 [JP] Japan
Apr. 4, 1990 [JP] Japan
Apr. 5, 1990 [JP] Japan
Apr. 12, 1990 [JP] Japan

[51] Int. Cl.
[52] U.S. Cl. 395/725; 364/240; 364/241.9; 364/DIG.

[58] Field of Search 370/85; 340/825; 364/200; 395/200, 250, 2

[56] References Cited

U.S. PATENT DOCUMENTS

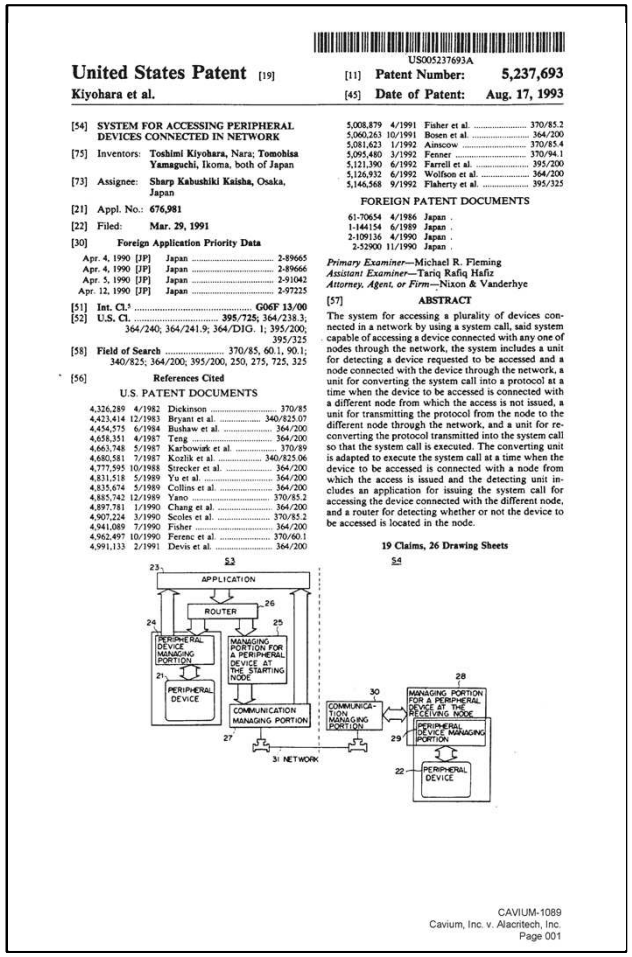
4,326,289 4/1982 Dickinson
4,423,414 12/1983 Bryant et al.
4,454,575 6/1984 Bushaw et al.
4,658,351 4/1987 Teng
4,663,748 5/1987 Karbowiak et al.
4,680,581 7/1987 Kozlik et al.
4,777,595 10/1988 Strecker et al.
4,831,518 5/1989 Yu et al.
4,835,674 5/1989 Collins et al.
4,885,742 12/1989 Yano
4,897,781 1/1990 Chang et al.
4,907,224 3/1990 Scoles et al.
4,941,089 7/1990 Fisher
4,962,497 10/1990 Ferenc et al.
4,991,133 2/1991 Devis et al.
5,008,879 4/1991 Fisher et al. 370/85.2

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001

FIG. 23 shows the procedure for receiving the packets from the transmission medium. When the communication LSI or hardware issues a receiving interruption, the header portion is cut out from the packet data and is stored in the header storage area for each layer (step SX8) and the data portion is stored in the data storage area for creating the management descriptor table DT (step SX9). After that, whether the communication

Ex. 1089 (Kiyohara) at 17:18-25; Paper 1 (Petition) at 44, 61; Ex.1003 (Horst Decl.) at A-13.

Kiyohara creates the descriptor table and the data pointer management table for new connections



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] **SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK**

[75] Inventors: **Toshimi Kiyohara**, Nara; **Tomohisa Yamaguchi**, Ikoma, both of Japan

[73] Assignee: **Sharp Kabushiki Kaisha**, Osaka, Japan

[21] Appl. No.: **676,981**

[22] Filed: **Mar. 29, 1991**

[30] **Foreign Application Priority Data**

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] **Int. Cl.**[5] **G06F 13/00**
[52] **U.S. Cl.** **395/725**; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] **Field of Search** 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,326,289	4/1982	Dickinson	370/85
4,423,414	12/1983	Bryant et al.	340/825.07
4,454,575	6/1984	Bushaw et al.	364/200
4,658,351	4/1987	Teng	364/200
4,663,748	5/1987	Karbowiak et al.	370/89
4,680,581	7/1987	Kozlik et al.	340/825.06
4,777,595	10/1988	Strecker et al.	364/200
4,831,518	5/1989	Yu et al.	364/200
4,835,674	5/1989	Collins et al.	364/200
4,885,742	12/1989	Yano	370/85.2
4,897,781	1/1990	Chang et al.	364/200
4,907,224	3/1990	Scoles et al.	370/85.2
4,941,089	7/1990	Fisher	364/200
4,962,497	10/1990	Ferenc et al.	370/60.1
4,991,133	2/1991	Devis et al.	364/200
5,008,879	4/1991	Fisher et al.	370/85.2
5,060,263	10/1991	Bosen et al.	364/200
5,081,623	1/1992	Ainscow	370/85.4
5,095,480	3/1992	Fenner	370/94.1
5,121,390	6/1992	Farrell et al.	395/200
5,126,932	6/1992	Wolfson et al.	364/200
5,146,568	9/1992	Flaherty et al.	395/325

FOREIGN PATENT DOCUMENTS

61-70654	4/1986	Japan .
1-144154	6/1989	Japan .
2-109136	4/1990	Japan .
2-52900	11/1990	Japan .

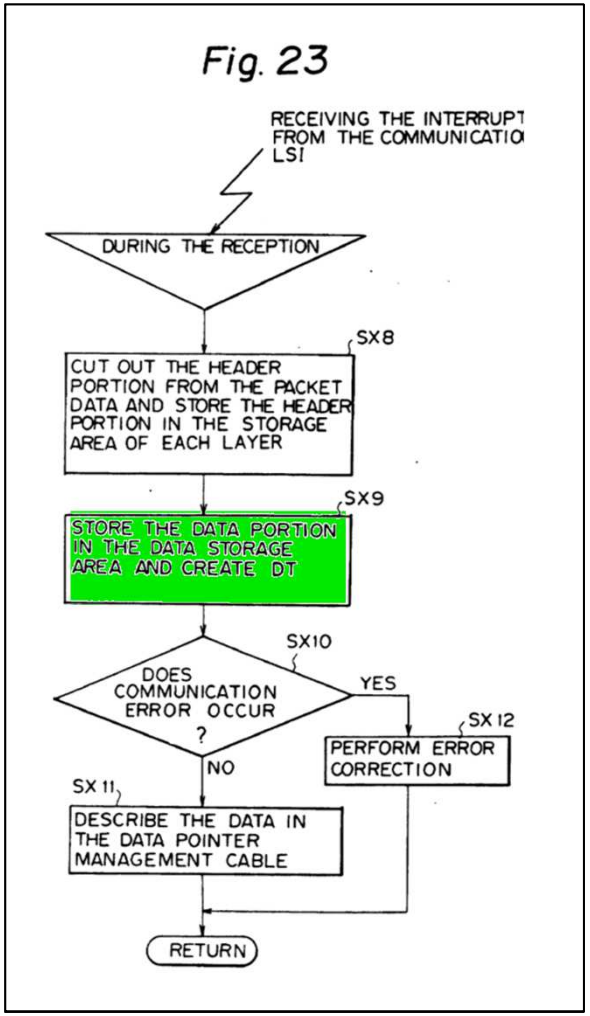
Primary Examiner—Michael R. Fleming
Assistant Examiner—Tariq Rafiq Hafiz
Attorney, Agent, or Firm—Nixon & Vanderhye

[57] **ABSTRACT**

The system for accessing a plurality of devices connected in a network by using a system call, said system capable of accessing a device connected with any one of nodes through the network, the system includes a unit for detecting a device requested to be accessed and a node connected with the device through the network, a unit for converting the system call into a protocol at a time when the device to be accessed is connected with a different node from which the access is not issued, a unit for transmitting the protocol from the node to the different node through the network, and a unit for re-converting the protocol transmitted into the system call so that the system call is executed. The converting unit is adapted to execute the system call at a time when the device to be accessed is connected with a node from which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

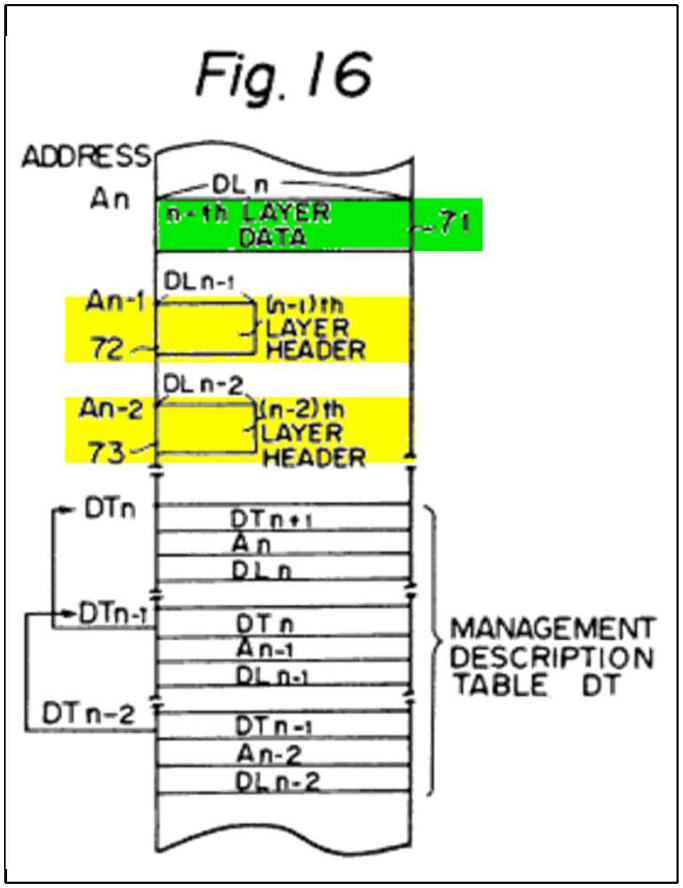
19 Claims, 26 Drawing Sheets

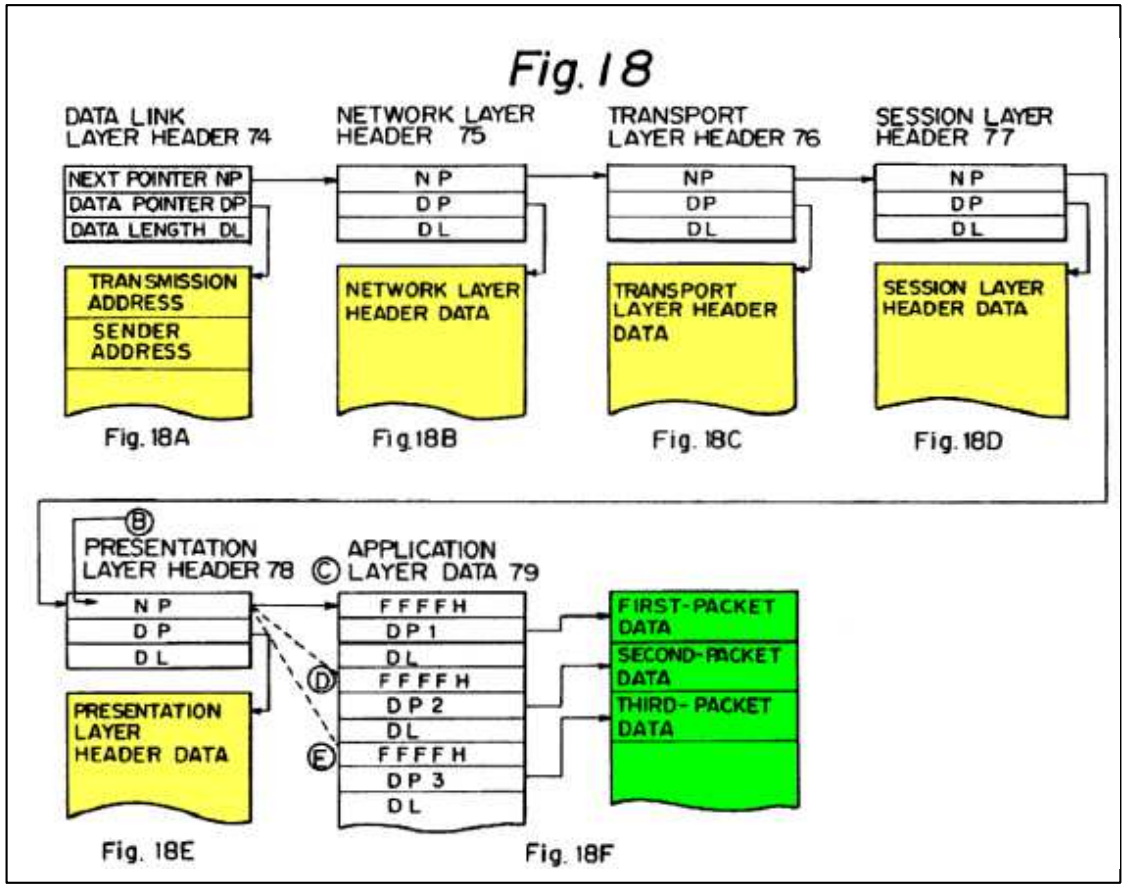
CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001



Ex. 1089 (Kiyohara) at Fig. 23, 17:18-25;
Paper 1 (Petition) at 39-42; Ex.1003 (Horst Decl.) at A-13.

Kiyohara discloses that the header portions and data portions are stored in different locations in memory

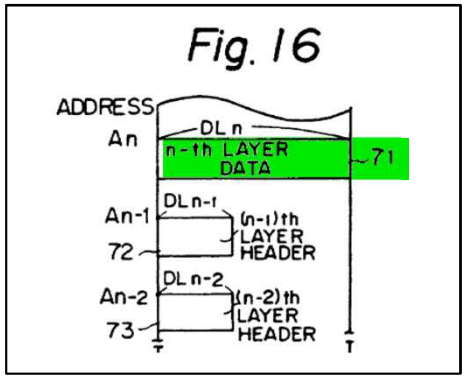




Ex. 1089 (Kiyohara) at Figs. 16, 18; Paper 1 (Petition) at 39-44;
Ex.1003 (Horst Decl.) at 54-58, A-13; Paper 29 (Reply) at 2, 5-7.

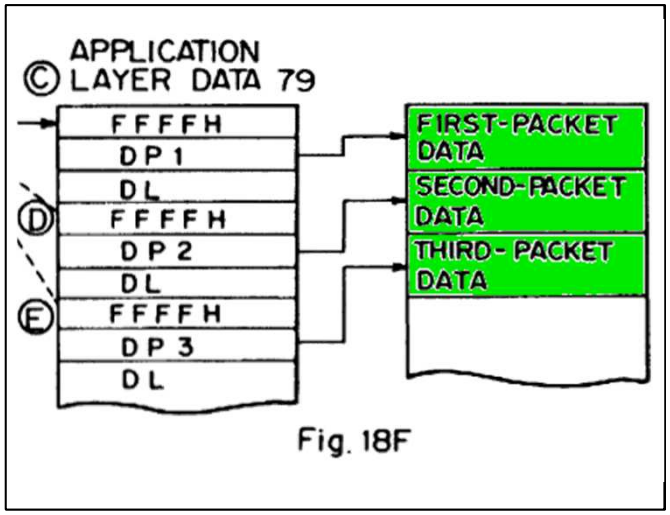
Kiyohara discloses that only the data portions of the packet are stored sequentially

The physical structure comprises n-th layer data 71 having a length of DLn to be written in sequence from an address An, a (n−1)th layer header 72 having a length of DLn-1 to be written at an address An-1, a (n−2)th layer header 73 having a length of DLn-2 to be written at an address An-2, and the like.

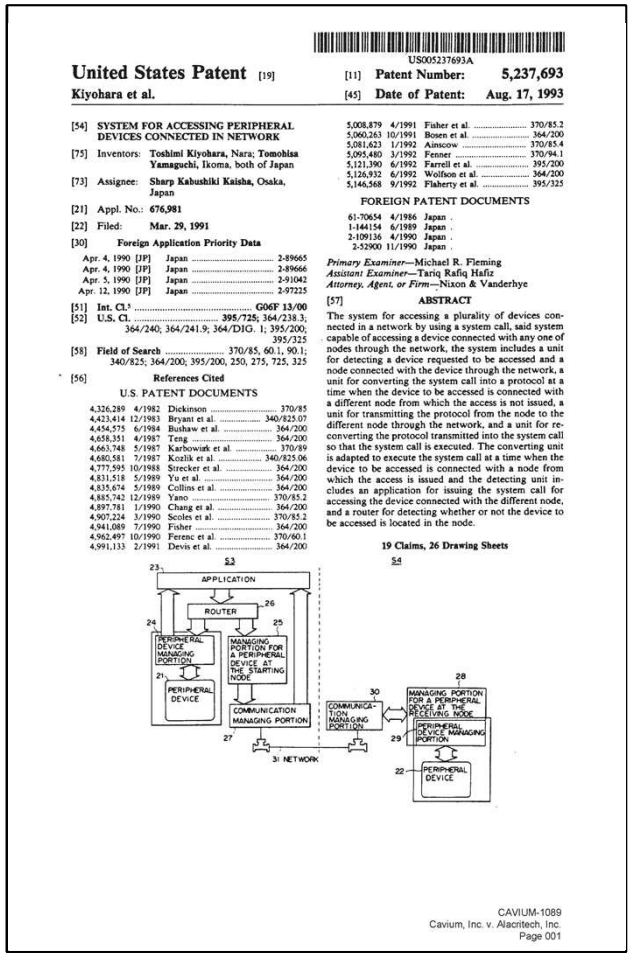


According to the present embodiment, therefore, when data is received, only the data is kept sequentially stored in the data storage area. It is thus unnecessary to

Ex. 1089 (Kiyohara) at Fig. 16, Fig. 18, 15:20-25, 17:32-44; Paper 1 (Petition) at 42-44; Ex.1003 (Horst Decl.) at 57-58; Paper 29 (Reply) at 4-6.



Kiyohara bypasses protocol processing for layers that have expected headers



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] **SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK**

[75] Inventors: **Toshimi Kiyohara**, Nara; **Tomohisa Yamaguchi**, Ikoma, both of Japan

[73] Assignee: **Sharp Kabushiki Kaisha**, Osaka, Japan

[21] Appl. No.: **676,981**

[22] Filed: **Mar. 29, 1991**

[30] **Foreign Application Priority Data**

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] **Int. Cl.**[5] **G06F 13/00**
[52] **U.S. Cl.** **395/725**; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] **Field of Search** 370/85, 60.1, 90.1; 340/825; 364/200, 395/200, 250, 275, 725, 325

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,326,289	4/1982	Dickinson	370/85
4,423,414	12/1983	Bryant et al.	340/825.07
4,454,575	6/1984	Bushaw et al.	364/200
4,658,351	4/1987	Teng	364/200
4,663,748	5/1987	Karbowiak et al.	370/89
4,680,581	7/1987	Kozlik et al.	340/825.06
4,777,595	10/1988	Strecker et al.	364/200
4,831,518	5/1989	Yu et al.	364/200
4,835,674	5/1989	Collins et al.	364/200
4,885,742	12/1989	Yano	370/85.2
4,897,781	1/1990	Chang et al.	364/200
4,907,224	3/1990	Scoles et al.	370/85.2
4,941,089	7/1990	Fisher	364/200
4,962,497	10/1990	Ferenc et al.	370/60.1
4,991,133	2/1991	Devis et al.	364/200
5,008,879	4/1991	Fisher et al.	370/85.2
5,060,263	10/1991	Bosen et al.	364/200
5,081,623	1/1992	Ainscow	370/85.4
5,095,480	3/1992	Fenner	370/94.1
5,121,390	6/1992	Farrell et al.	395/200
5,126,932	6/1992	Wolfson et al.	364/200
5,146,568	9/1992	Flaherty et al.	395/325

FOREIGN PATENT DOCUMENTS

61-70654	4/1986	Japan .
1-144154	6/1989	Japan .
2-109136	4/1990	Japan .
2-52900	11/1990	Japan .

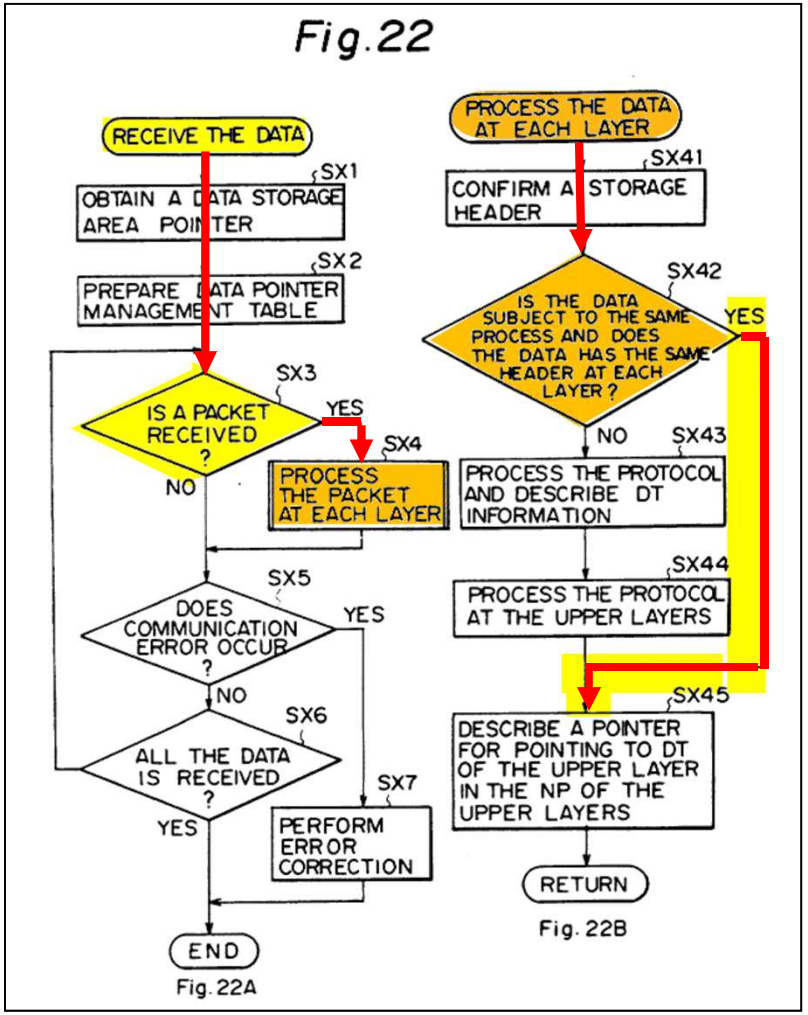
Primary Examiner—Michael R. Fleming
Assistant Examiner—Tariq Rafiq Hafiz
Attorney, Agent, or Firm—Nixon & Vanderhye

[57] **ABSTRACT**

The system for accessing a plurality of devices connected in a network by using a system call, said system capable of accessing a device connected with any one of nodes through the network, the system includes a unit for detecting a device requested to be accessed and a node connected with the device through the network, a unit for converting the system call into a protocol at a time when the device to be accessed is connected with a different node from which the access is not issued, a unit for transmitting the protocol from the node to the different node through the network, and a unit for re-converting the protocol transmitted into the system call so that the system call is executed. The converting unit is adapted to execute the system call at a time when the device to be accessed is connected with a node from which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

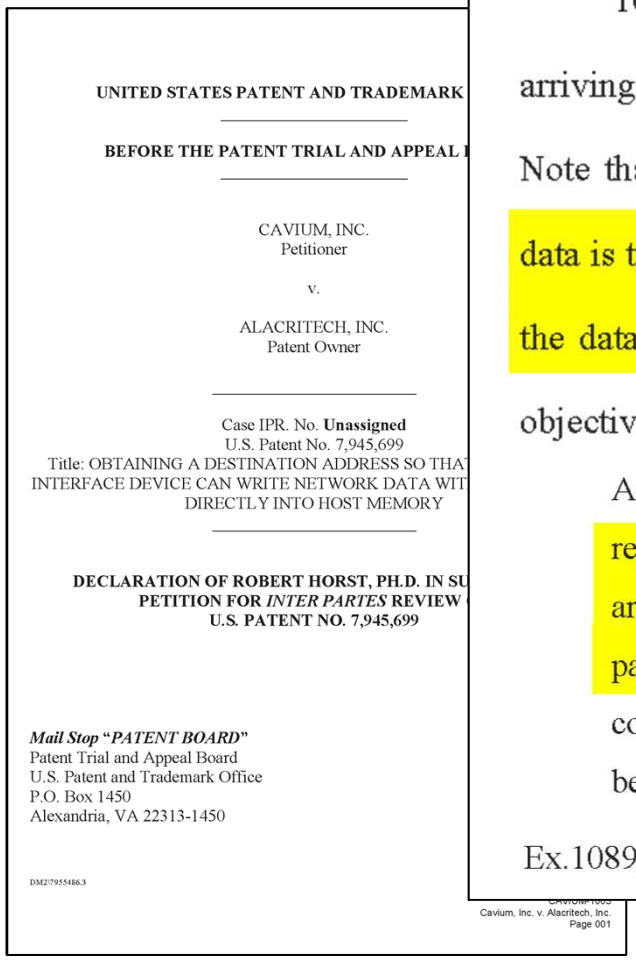
19 Claims, 26 Drawing Sheets

CAVIUM-1089
Cavium, Inc. v. Alacritech, Inc.
Page 001



Ex. 1089 (Kiyohara) at Fig. 22; Paper 1 (Petition) at 39-42, 44; Ex.1003 (Horst Decl.) at 57-59.

After the data pointer management table is created, the data from each packet is moved to the data storage area



UNITED STATES PATENT AND TRADEMARK

BEFORE THE PATENT TRIAL AND APPEAL

CAVIUM, INC.
Petitioner

v.

ALACRITECH, INC.
Patent Owner

Case IPR. No. **Unassigned**
U.S. Patent No. 7,945,699
Title: OBTAINING A DESTINATION ADDRESS SO THA
INTERFACE DEVICE CAN WRITE NETWORK DATA WIT
DIRECTLY INTO HOST MEMORY

DECLARATION OF ROBERT HORST, PH.D. IN SU
PETITION FOR *INTER PARTES* REVIEW
U.S. PATENT NO. 7,945,699

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

DM217955486.3

Cavium, Inc. v. Alacritech, Inc.
Page 001

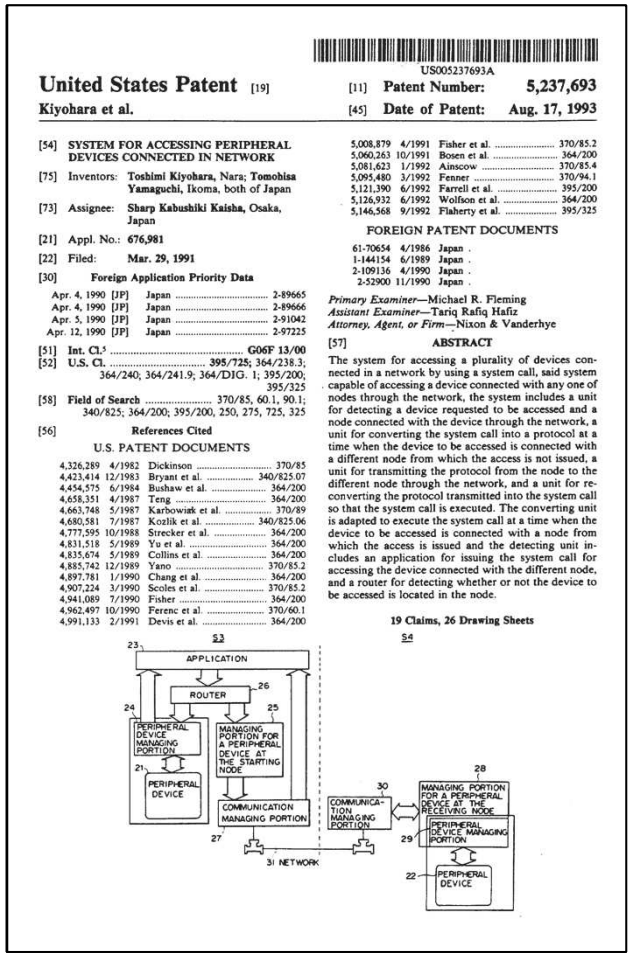
108. When receiving a packet, once the linked list of descriptors is set up, arriving packets for the same connection transfer data directly to the data location. Note that the locations of header information and data are different and received data is transferred to the application layer data area without transferring headers to the data area. This avoidance of data and header copies between layers is a key objective of Kiyohara:

> According to the present embodiment, therefore, when data is received, only the data is kept sequentially stored in the data storage area.... Further, since only the data portion is extracted out of the packet, it is possible to reduce the frequency of unnecessary data copying between the layers, thereby allowing the communication to be realized at high-speed.

Ex.1089, Kiyohara at 17:32-45.

Ex. 1003 (Horst Decl) at 57-58;
Paper 1 (Petition) at 44.

Kiyohara performs protocol processing if the headers are not the expected headers



US005237693A

United States Patent [19]
Kiyohara et al.

[11] **Patent Number: 5,237,693**
[45] **Date of Patent: Aug. 17, 1993**

[54] **SYSTEM FOR ACCESSING PERIPHERAL DEVICES CONNECTED IN NETWORK**

[75] Inventors: **Toshimi Kiyohara**, Nara; **Tomobisa Yamaguchi**, Ikoma, both of Japan

[73] Assignee: **Sharp Kabushiki Kaisha**, Osaka, Japan

[21] Appl. No.: **676,981**

[22] Filed: **Mar. 29, 1991**

[30] **Foreign Application Priority Data**

Apr. 4, 1990 [JP] Japan 2-89665
Apr. 4, 1990 [JP] Japan 2-89666
Apr. 5, 1990 [JP] Japan 2-91042
Apr. 12, 1990 [JP] Japan 2-97225

[51] **Int. Cl.**[5] **G06F 13/00**
[52] **U.S. Cl.** **395/725**; 364/238.3; 364/240; 364/241.9; 364/DIG. 1; 395/200; 395/325
[58] **Field of Search** 370/85, 60.1, 90.1; 340/825; 364/200; 395/200, 250, 275, 725, 325

[56] **References Cited**
U.S. PATENT DOCUMENTS

4,326,289	4/1982	Dickinson	370/85
4,423,414	12/1983	Bryant et al.	340/825.07
4,454,575	6/1984	Bushaw et al.	364/200
4,658,351	4/1987	Teng	364/200
4,663,748	5/1987	Karbowiak et al.	370/89
4,680,581	7/1987	Kozlik et al.	340/825.06
4,777,595	10/1988	Strecker et al.	364/200
4,831,518	5/1989	Yu et al.	364/200
4,835,674	5/1989	Collins et al.	364/200
4,885,742	12/1989	Yano	370/85.2
4,897,781	1/1990	Chang et al.	364/200
4,907,224	3/1990	Scoles et al.	370/85.2
4,941,089	7/1990	Fisher	364/200
4,962,497	10/1990	Ferenc et al.	370/60.1
4,991,133	2/1991	Devis et al.	364/200
5,008,879	4/1991	Fisher et al.	370/85.2
5,060,263	10/1991	Bosen et al.	364/200
5,081,623	1/1992	Ainscow	370/85.4
5,095,480	3/1992	Fenner	370/94.1
5,121,390	6/1992	Farrell et al.	395/200
5,126,932	6/1992	Wolfson et al.	364/200
5,146,568	9/1992	Flaherty et al.	395/325

FOREIGN PATENT DOCUMENTS

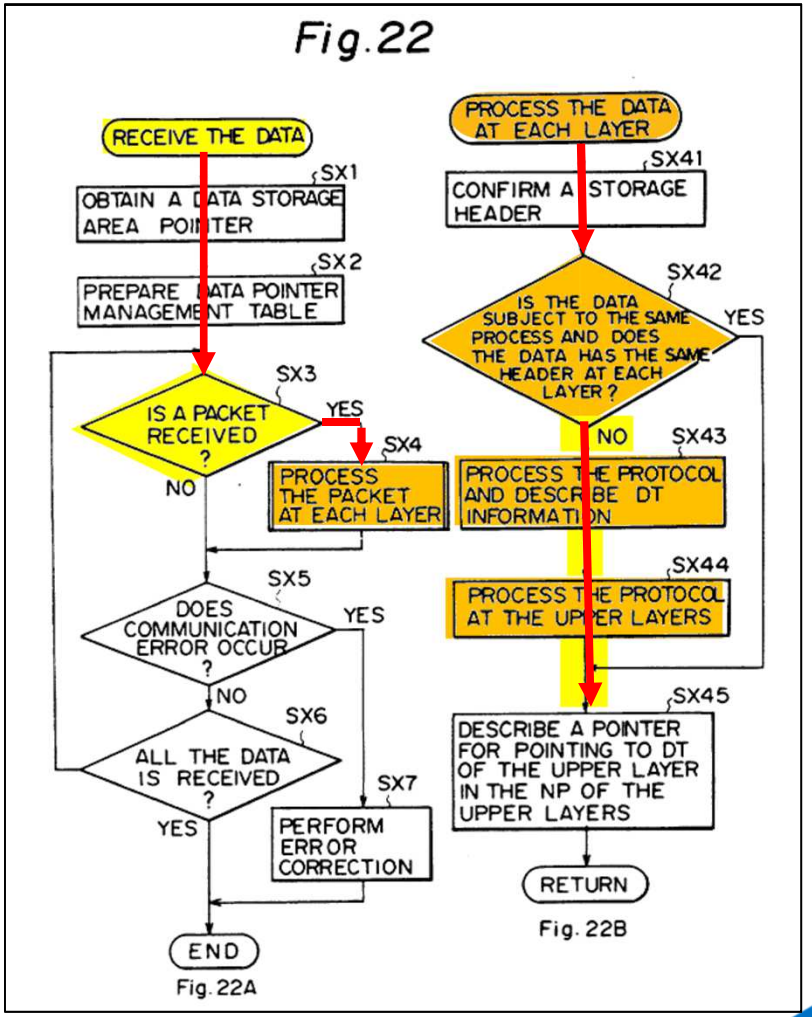
61-70654	4/1986	Japan .
1-144154	6/1989	Japan .
2-109136	4/1990	Japan .
2-52900	11/1990	Japan .

Primary Examiner—Michael R. Fleming
Assistant Examiner—Tariq Rafiq Hafiz
Attorney, Agent, or Firm—Nixon & Vanderhye

[57] **ABSTRACT**

The system for accessing a plurality of devices connected in a network by using a system call, said system capable of accessing a device connected with any one of nodes through the network, the system includes a unit for detecting a device requested to be accessed and a node connected with the device through the network, a unit for converting the system call into a protocol at a time when the device to be accessed is connected with a different node from which the access is not issued, a unit for transmitting the protocol from the node to the different node through the network, and a unit for reconverting the protocol transmitted into the system call so that the system call is executed. The converting unit is adapted to execute the system call at a time when the device to be accessed is connected with a node from which the access is issued and the detecting unit includes an application for issuing the system call for accessing the device connected with the different node, and a router for detecting whether or not the device to be accessed is located in the node.

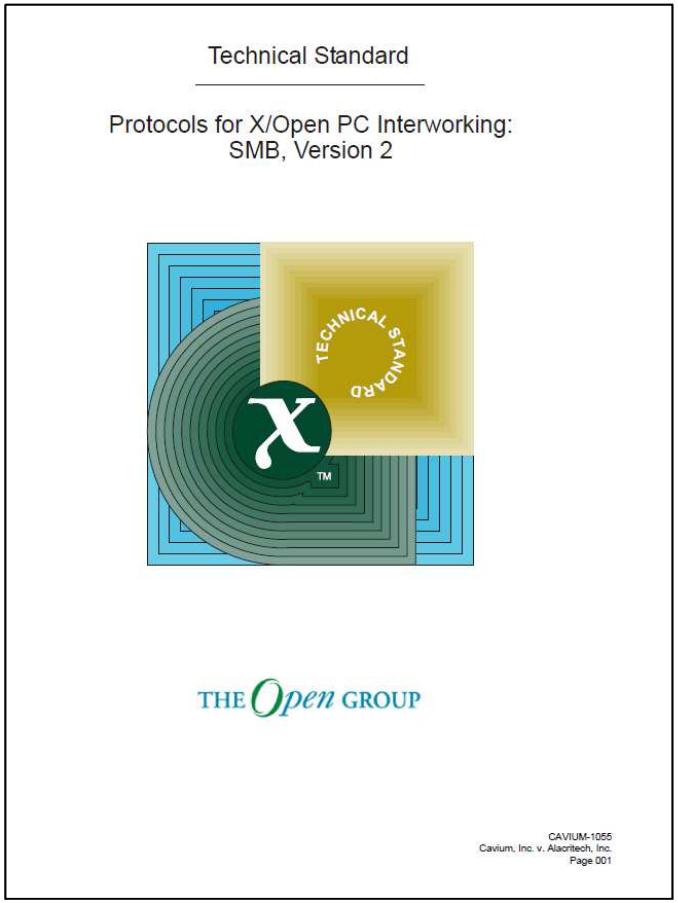
19 Claims, 26 Drawing Sheets



Ex. 1089 (Kiyohara) at Fig. 22; Paper 1 (Petition) at 39-42, 44; Ex.1003 (Horst Decl.) at 58-59.

Tutorial on Prior Art References

- **Protocols for X/Open PC Interworking: SMB, Version 2 (“SMB”) (Ex. 1055)**


Technical Standard

Protocols for X/Open PC Interworking:
SMB, Version 2

THE Open GROUP

CAVIUM-1055
Cavium, Inc. v. Alacritech, Inc.
Page 001

SMB was the industry standard protocol for communicating with Microsoft PCs

Technical Standard

SMB

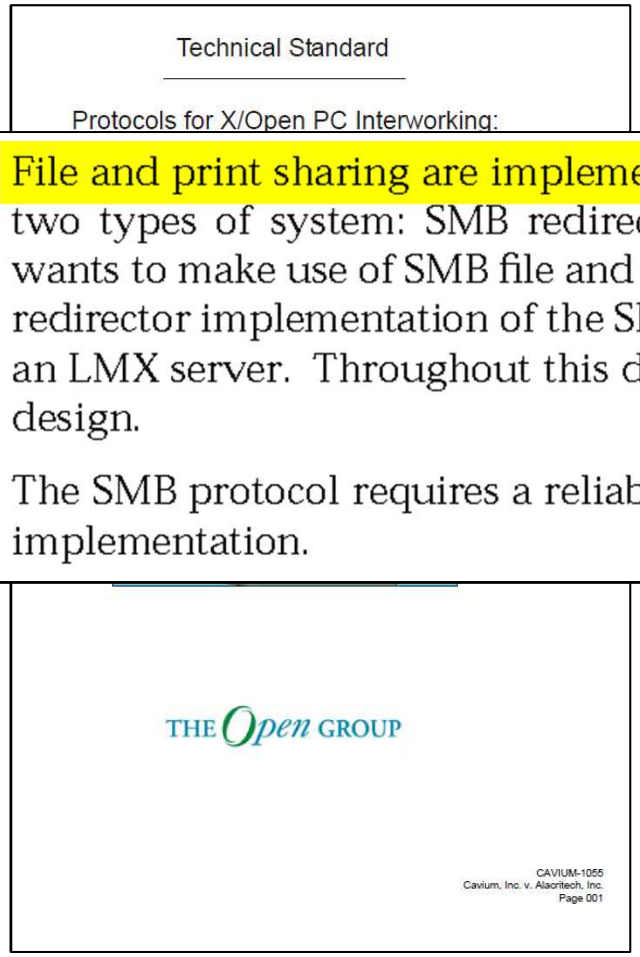
(Server Message Block) A protocol which allows a set of computers to access shared resources as if they were local. The core protocol was developed by Microsoft Corporation and Intel, and the extended protocols were developed by Microsoft Corporation.

When connecting personal computers and X/Open-compliant systems via standard transport protocols, there appear to be two possibly overlapping but distinct market segments. In the first one, personal computers are added to existing networks of X/Open-compliant systems which already have a distributed file system, the most widely-adopted one being the Network File System originally designed by Sun Microsystems. In the second one, X/Open-compliant servers are added to LANs consisting primarily of personal computers. For personal computers running under DOS or OS/2 operating systems, which is the vast majority, the generally accepted non-proprietary protocol is the Server Message Block from Microsoft Corporation.

CAVIUM-1055
Cavium, Inc. v. Alacritech, Inc.
Page 001

Ex. 1055.014, .526 (SMB); Paper 1 (Petition) at 49;
Ex.1003 (Horst Decl.) at 59-60.

SMB provides file and print sharing


Technical Standard

Protocols for X/Open PC Interworking:

THE Open GROUP

CAVIUM-1055
Cavium, Inc. v. Alacritech, Inc.
Page 001

File and print sharing are implemented using the SMB protocol. This protocol is used between two types of system: SMB redirectors and LMX servers. When a user on an SMB redirector wants to make use of SMB file and print services available in the network the user needs an SMB redirector implementation of the SMB protocol. Upon request the SMB redirector will connect to an LMX server. Throughout this document the term LMX server does not imply any particular design.

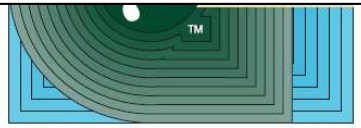
The SMB protocol requires a reliable connection-oriented virtual circuit provided by a NetBIOS implementation.

Ex. 1055.022 (SMB); Paper 1 (Petition) at 46-49;
Ex.1003 (Horst Decl.) at 59-60.

SMB supported TCP using NetBIOS

Technical Standard

Protocols for X/Open PC Interworking:
SMB, Version 2



THE Open GROUP

CAVIUM-1055
Cavium, Inc. v. Alacritech, Inc.
Page 001

The SMB protocol makes use of a NetBIOS transport facility. NetBIOS defines a set of network transport facilities. The interface is outside the scope of this document. The NetBIOS functions can be implemented over a variety of transport protocols, however within this document only the mapping of NetBIOS over TCP and UDP (see Appendices F and G) and NetBIOS over ISO transport services (see Appendix E on page 281) are considered.

Ex. 1055.032 (SMB); Paper 1 (Petition) at 47;
Ex.1003 (Horst Decl.) at 61-62.

The *smb_dsize* header field in the response to an SMB read command indicates the amount of data returned

UNITED STATES PATENT AND TRADEMARK OFF

BEFORE THE PATENT TRIAL AND APPEAL BOA

CAVIUM, INC.
Petitioner

v.

ALACRITECH, INC.
Patent Owner

Case IPR. No. **Unassigned**
U.S. Patent No. 7,945,699
Title: OBTAINING A DESTINATION ADDRESS SO THAT A
INTERFACE DEVICE CAN WRITE NETWORK DATA WITHOU
DIRECTLY INTO HOST MEMORY

DECLARATION OF ROBERT HORST, PH.D. IN SUPPO
PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 7,945,699

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

DM217955486.3

CAVIUM-1003
Cavium, Inc. v. Alacritech, Inc.
Page 001

117. The left column shows the packet format for sending a request, and the right column the format for receiving the response. In the response, *smb_data* is the variable length data field returned, and everything before that is the header. The header includes the length of the data which is returned in *smb_dsize*.

smb_dsize	This unsigned 16-bit field contains the number of bytes of data actually read and returned in this response.
smb_doff	This unsigned 16-bit field indicates the offset from the SMB header to the start of the returned data, in bytes. This permits variable-sized padding.
smb_rsvd	These two 16-bit and four 16-bit fields are padding that force the *SMBreadX* response to be the same size as the *SMBwriteX* request. They must be zero.
smb_pad	This field is between zero and three 8-bit fields in length, as governed by the *smb_doff* field. It may be used by an LMX server to pad the size of the *SMBreadX* response out to a 16-bit or 32-bit boundary which provides the best performance.
smb_data	The actual data read from the file.

Ex.1055, SMB at .179.

Ex. 1003 (Horst Decl.) at 64;
Paper 1 (Petition) at 57-59.

The Samba application used the *smb_dsize* header field to determine the memory storage location

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CAVIUM, INC.
Petitioner

v.

ALACRITECH, INC.
Patent Owner

Case IPR. No. **Unassigned**
U.S. Patent No. 7,945,699
Title: OBTAINING A DESTINATION ADDRESS SO THAT A NETWORK INTERFACE DEVICE CAN WRITE NETWORK DATA WITHOUT HEADERS DIRECTLY INTO HOST MEMORY

DECLARATION OF ROBERT HORST, PH.D. IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,945,699

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

DM2\7955486.3

CAVIUM-1003
Cavium, Inc. v. Alacritech, Inc.
Page 001

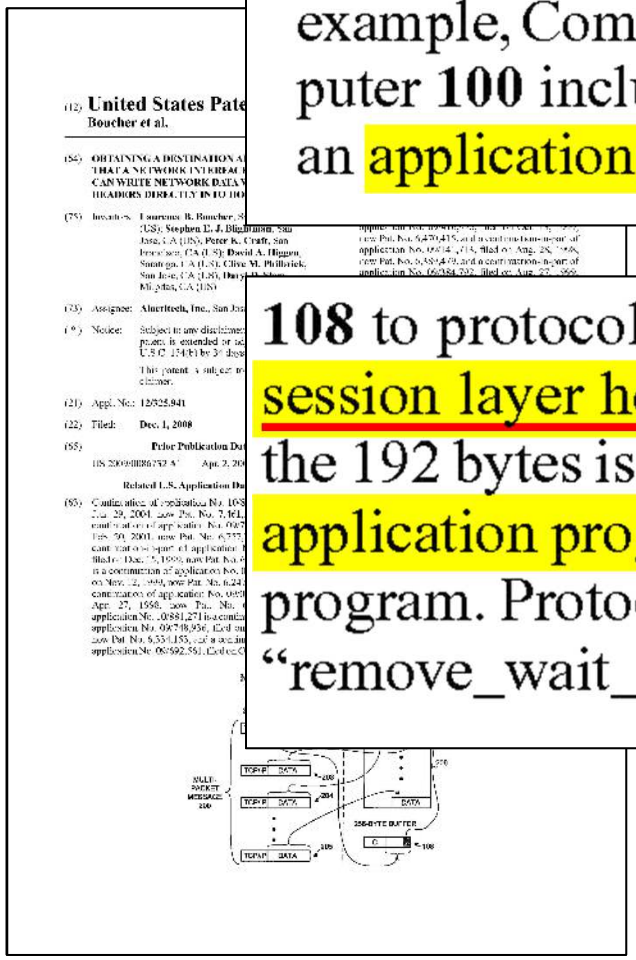
implementation, the following code from a 1996 Samba release is an excerpt from the *SMBreadX* command. These excerpts from the client.c file show part of the code that uses *SMBreadX* to determine the memory locations for buffering data from a remote file before writing that part of the remote file to a local file.

```
/* we might have got some data from a chained readX */
if (SVAL(inbuf,smb_vwv0) == SMBreadX)
  {
    p = (smb_base(inbuf)+SVAL(inbuf,smb_vwv1)) - smb_wct;
    datalen = SVAL(p,smb_vwv5);
    dataptr = smb_base(inbuf) + SVAL(p,smb_vwv6);
  }

    /* 3 possible read types. readbraw if a large block is required.
   readX + close if not much left and read if neither is supported */

    /* we might have already read s
    if (dataptr && datalen>0)
  method=3;

  case 3:
    /* we've already read some data
    break;

    datalen = SVAL(inbuf,smb_vwv0);
    dataptr = smb_buf(inbuf) + 3;
    break;
  }

    if (writefile(handle,dataptr,da
```

To SMB redirector	
Field Name	Field Value
smb_com	*SMBreadX*
smb_wct	12
smb_vwv[0]	*smb_com2*
smb_vwv[1]	*smb_off2*
smb_vwv[2]	*smb_remaining*
smb_vwv[3-4]	*smb_rsvd*
smb_vwv[5]	*smb_dsize*
smb_vwv[6]	*smb_doff*
smb_vwv[7-10]	*smb_rsvd*
smb_bcc	(data length + pad)
smb_buf[ ]	*smb_pad*
	smb_data

Ex. 1003 (Horst Decl.) at 64-66, 113; Paper 1 (Petition) at 57-60.

The 699 patent's preferred embodiment uses Samba



example, Compaq Proliant). Software executing on host computer **100** includes: 1) a Linux operating system **103**, and 2) an application program **104** by the name of "Samba". Oper-

108 to protocol stack **107**. The first part of this 192 bytes is session layer header information, whereas the remainder of the 192 bytes is session layer data. Protocol stack **107** notifies application program **104** that there is data for the application program. Protocol stack **107** does this by making a call to the "remove_wait_queue" routine.

Ex. 1001 (699 Patent) at 3:53-55, 5:5-10; Paper 1 (Petition) at 29-30;
Ex.1003 (Horst Decl.) at 47-48;
Paper 29 (Reply) at 14-15.