

Intel Corp. v. Alacritech, Inc. IPR2017-01391, -01392, -01393, -01405, -01406, -01409, -01410

September 13, 2018



Common-Interest Privileged

(1) 036 Patent (IPR2017-01391)

1.	Motivation to Combine Erickson and Tanenbaum96	Slides 9-54
2.	Prior Art Discloses 036 Limitations	Slides 55-77
3.	Motion to Amend 036 Patent	Slides 78-96
(2) 0	72 Patent (IPR2017-01406)	
1.	Motivation to Combine Erickson and Tanenbaum96	Slide 100
2.	Prior Art Discloses 072 Limitations	Slides 101-118
3.	Motion to Amend 072 Patent	Slides 119-137
(3) 2	41 Patent (IPR2017-01392)	
1.	Motivation to Combine Erickson, Tanenbaum96 (and Alteon)	Slides 141-149
2.	Prior Art Discloses 241 Limitations	Slides 150-187



(3)	241	Patent	(IPR2017-01392)	(Continued)
-----	-----	--------	-----------------	-------------

3.	Alteon is Prior Art	Slides 188-193		
4.	Motion to Amend 241 Patent	Slides 194-206		
4) 880 Patent (IPR2017-01409, -1410)				
1.	Motivation to Combine Thia, Tanenbaum (and Nahum)	Slides 209-228		
2.	Thia and Nahum are Enabling	Slides 229-232		
3.	Prior art Discloses 880 Limitations	Slides 233-266		
4.	Motions to Amend 880 Patents	Slides 267-288		

(5) 205 Patent (IRP2017-01405)

- 1. Thia is Enabling Prior Art
- 2. Thia Teaches Claimed Processing

Slides 291-296 Slides 297-318



(5) 205 Patent (IPR2017-01405) (Continued)

- 3. Prior Art Discloses Challenged Claims
- 4. Motivation to Combine
- 5. Supplemental Briefing (Claims 31-33)
- 6. Motions to Amend 205 Patent
- (6) 104 Patent (IPR2017-01393)
 - 1. Prior Art Discloses 104 Limitations
 - 2. Supplemental Briefing (Claim 22)

(7) Common Issues

1. Secondary Considerations (IPR2017-01391, -01392, 01393, -01405, -01406, -01409, -01410)

Slides 319-321 Slides 322-333 Slides 334-341 Slides 342-348

Slides 352-379 Slides 380-390

Slides 391-394



(7) Common Issues (Continued)

- a) Real Party in Interest (IPR2017-01391, -01392, -01393, Slides 395-399 - 01405, -01406, -01409, -01410)
- b) Tanenbaum was Publicly Accessible (IPR2017-01391, Slides 400-413 01392, -01406, -01409, -10410)

U.S. Patent No. 7,237,036 (036 Patent)

IPR2017-1391 (Intel) IPR2018-0371 (Dell) IPR2017-1718 (Cavium) IPR2018-0327 (Wistron)

*All citations herein are to the IPR2017-01391 case unless otherwise noted.

(intel)

036 Patent: Instituted Grounds

• Erickson in view of Tanenbaum96

• 036 Patent: Claims 1, 2-7



036 Patent: Disputes

1. A POSA would have been motivated to combine Tanenbaum96 with Erickson

2. Erickson in view of Tanenbaum96 discloses the limitations of claims 1-7 of the 036 Patent

3. Motion to Amend 036 Patent should be denied

(intel) 8

036 Patent: Disputes

- 1. A POSA would have been motivated to combine Tanenbaum96 with Erickson
 - a) <u>A POSA would have naturally looked to Tanenbaum96</u> when implementing the TCP functionality disclosed in <u>Erickson</u>
 - b) Tanenbaum96 does not teach away from the invention
 - A POSA would have a reasonable expectation of success using Tanenbaum96 to implement Erickson's TCP functionality
 - d) Dr. Horst's 2001 Article shows that "conventional wisdom" was to offload TCP

Erickson: Use of fast and slow applications



Erickson: Adapter offloads protocol processing for fast applications



with the present invention. A traditional slow application 306 uses normal streams processing 308 to send information

interface 322. With the present invention, fast user applications 302 and 304 directly use a setup driver 312 to initialize the physical hardware registers 320, then send the information directly through the I/O device adapter 314 to the commodity interface 322 via virtual hardware 316 and 318. Thus, the overhead of the normal streams processing 308 and pass-through driver 310 are eliminated with the use of the virtual hardware 316 and 318 of the present invention, and fast applications 302 and 304 are able to send and receive information more quickly than slow application 306.

Paper 2 (036 Petition) at 40-41; Paper 1 (072 Petition) at 35-37; Ex. 1003.065-.066 (036 Horst Decl.); Ex. 1003.067-.068, .079-.084 (072 Horst Decl.); Ex. 1005 (Erickson) at 4:53-5:3.

(intel)

Erickson: Fast receive and transmit



Paper 2 (036 Petition) at 44-45; Paper 1 (072 Petition) at 35-37; Ex. 1003.077, .079-.084 (072 Horst Decl.); Ex. 1005 (Erickson) at 5:6-14, Fig. 4.

intel 12

Erickson: Adapter stores protocol scripts and data for moving data

Un	ited S	States Patent [19]	[11]	Patent Number:	5,768,61
Eric	kson et	al.	[45]	Date of Patent:	Jun. 16, 19
[54]	METHOI OF ACTI COMPUT VALUES SUB POR	D FOR PERFORMING SEQUEN IONS IN DEVICE CONNECTED TER IN RESPONSE TO SPECIF BEING WRITTEN INTO SNOO RTIONS OF ADDRESS SPACE	TO and Shin IED 1988. C OPED Californ	ASH Local Kernal Structure - Yuan Tzou, Report No. U(computer Science Division ia. Berkeley 94720.	" by David P. Anders (B/CSD 88/463, Nov. (EECS), University
[75]	Inventors:	Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keit Muller; Curtis H. Stehley, both Diego, all of Calif.	"A User municati of San National Box 20	s' Guide to PICL—A Porta ion Library" By G.A. Ge Laboratory, Mathematical 09. Bidg, 9207–A. Oak R	ible Instrumented Co ist et. al., Oak Rid Sciences Section. P. Lidge, TN 37831-80
[73]	Assignce:	NCR Corporation. Dayton. Ohio	(Aug. 19	790).	d Malana" Bu Casia
[21]	Appl. No.: Filed:	: 577,678	Stunkel. New Yo	et. al., IBM Research Divis rk (Sep. 22, 1993).	of Vulcan" By Craig sion, Yorktown Heigh
[51] [52] [58]	Int. CL ⁶ U.S. Cl Field of S	G06F	7 15/02 "MPI-F 95/829 by Hub 1. 823. Research 9. 500.	An MPI Prototype Implem ertus Franke et. al., pub. h Center, Yorktown Heights	nentation on IBM SP by IBM, T.J. Wats s, New York 10598.
[56]	U.	References Cited S. PATENT DOCUMENTS	473 Primary Attorney Welter 8	Examiner—Moustafa M. b Agent, or Firm—Merchan & Schmidt	deky at. Gould. Smith. Ede
4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5	589,063 5 777,589 10 016,161 5 016,166 5 127,098 6 280,587 1 420,987 5 548,778 8 553,244 9 642,481 6 671,442 9 FORE	\$1986 Stab et al. \$1988 Boeners et al. \$1998 Boeners et al. \$1999 Ya Loo et al. \$1999 Ya Loo et al. \$1999 Koensthal et al. \$1999 Koensthal et al. \$1998 Koensthal et al. \$1998 Koensthal et al. \$1998 Koensthal et al. \$1999 Montows et al. \$1999 <	995/828 [57] 995/823 [57] 995/823 A methol 995/823 address 995/823 memory 995/820 address 995/823 prises vi 995/824 the UO devi 995/825 the VO devi 995/824 the UO d	ABSTRACT do of controlling an input/o a computer to facilitate fas space for the I/O device is of the computer, wherein a trual registers that are used ce. In essence, control regi- evice are mapped into the vi al address space is backte emory on the I/O device. The	utput (I/O) device co st I/O data transfers. J s created in the virtu the address space cool t to direc- sters and trual ade of by o
The	551148 7 (Performand	7/1993 European Pat. Off OTHER PUBLICATIONS ce. of Message-Passing Using Res	detects v sequence stricted program	rites to the address space. A e of actions can be triggere ming specified values into t	the data
Virtua David 21(3).	P. Anderso 251–267 (Remapping", by Shin-Yuan Tzo m. in Software-Practice & Experient (Mar. 1991).	ou and mapped ce, vol.	19 Claims, 7 Drawin	we Sheet aC
		TROUGEN MADE	POTECAL NEMORY DNI-O ADMPTER	UNIVE NEWSEN VIETUR NEWSEN IN REPORT	pr
		84			W



-512 MAIN MEMORY SOFTWARE 516 REGISTER ENDPOINT TABLE INDEXED BY 508 J PROTOCOL HARDWARE R / W APPLICATION ID SCRIPTS APPLICATION RO APPLICATION R / W - 514 HARDWARE RO HARDWARE REGISTER ENDPOINT 504 PROTOCOL DATA . 510 506 BUFFER PHYSICAL ADDRESS OS DRIVER R / W POOL BUFFER MAP 518 HARDWARE RO - 502

ADAPTER MEMORY

sses. Each entry within the endpoint table 514 points to rious protocol data 518 in the memory 512 in order to commodate multiple communication protocols, as well as eviously defined protocol scripts 516 in the memory 512, nich indicate how data or information is to be transferred from the memory 512 of the I/O device adapter to the portions of main memory 502 associated with a user process.

Paper 2 (036 Petition) at 41-42; Paper 1 (072 Petition) at 37; Ex. 1005 (Erickson) at 5:61-67.

Erickson: Adapter executes the scripts



Jun. 16, 1998

[11] Patent Number: [45] Date of Patent:

United States Patent [19] Erickson et al.

[54] METHOD FOR PERFORMING SEQUENCE OF ACTIONS IN DEVICE CONNECTED TO COMPUTER IN RESPONSE TO SPECIFIED

VALUES BEING WRITTEN INTO SNOOPED SUB PORTIONS OF ADDRESS SPACE ntors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of San Disco all of Colif

Diego, all of Calif. [73] Assignce: NCR Corporation. Dayton, Ohio

[21] Appl. No.: 577,678 [22] Filed: Dec. 21, 1995

[51] Int. CL⁶ _ G06F 15/02 [52] U.S. CL [58] Field of Search

arch _____ 395/821. 823. 395/829, 832, 846, 882, 284, 309, 500. 473

References Cited U.S. PATENT DOCUMENTS

4.589,063 5/1986 Shah et al 4.777,589 10/1988 Boetsner et al 5.016,161 5/1991 Van Loo et al 5.016,163 5/1991 Van Loo et al 5.127,098 6/1992 Rosensthal et al ... 5.420,087 5/1994 Shimodhine et al ... 5.420,087 5/1994 Shimodhine et al ... 395/82 395/828 395/823 395/678 395/678 711/202 395/880 395/830 5,420,987 5,548,778 5,42,987 5/1995 Reid et al. 5,548,778 8/1996 Hirayama 5,533,244 9/1996 Norcross et al. ... 5,642,481 6/1997 Pedrizetti 5,671,442 9/1997 Feeney et al. 395/823 395/280 395/185.01 _____395/834

FOREIGN PATENT DOCUMENTS 551148 7/1993 European Pat. Off.

OTHER PUBLICATIONS

"The Performance of Message-Passing Using Restricted Virtual Memory Remapping", by Shim-Yuan Tzou and David P. Anderson, in Software-Practice & Experience, vol. 21(3), 251-267 (Mar. 1991).

KRUSER PROC SECURE VEW

prises virtual registers that are used to directly control the I/O device. In essence, control registers and/or memory of the I/O device are mapped into the virtual address space, and the virtual address space is backed by control registers and/or memory on the I/O device. Thereafter, the I/O device detects writes to the address space. As a result, a pre-defined sequence of actions can be triggered in the I/O device by mapped virtual address space. 19 Claims, 7 Drawing Sheets

programming specified values into the data written into the

memory of the computer, wherein the address space con



Paper 2 (036 Petition) at 65; Paper 41 (036 Reply) at 15; Paper 1 (072 Petition) at 40-41; Ex. 1003.094 (072 Horst Decl.); Ex. 1005 (Erickson) at 4:18-23.

A script is prepared by the operating system for the I/O device adapter to execute each time the specific user process programs its specific virtual hardware. The user process is given a virtual address in the user process' address space that allows the user process very specific access capabilities to the I/O device adapter.

intel)

Erickson: A pre-negotiated template passed to the script on the adapter

-702

FIG. 7

(6 bytes)

(6 bytes)

00 Service Type

Total Length

Datagram Id

Time-to-Live = 0x40

IP Header Checksum

UDP Length

UDP Checksum

Target Ethernet Address

01

06 07

08

0a

0b

0c

08

45

40

00

40

11

80

01

c0

07

80

01

c0

08

00

07

30

18

00

Hex Dec 0 0

> 1 1

2 2 3 3 4 4

5 5

6 6

7 7

8 8 9 9

а 11 b

С

d 13

е

f 15

10 16

11 17 12 18

13 19

14 20 15 21

16 22

17 23

18 24

1b 27

1d 29

1e

1f 31

20 32

21 33

22 23 34 UDP

24 36

25 37

26 38

27 39 28 40

29 41

25 26 19 1a

28 1c

30

35

10

12

14

Ethernet

Header

(14 bytes)

704

Header

(20 bytes)

706

Header

(8 bytes)

708

The script that executes the above function provides the Source Ethernet Address USERDATA ADDRESS and USERDATA LENGTH which the user process programmed into the adapter's Protocol Type (0x0800 = IF memory. This information quite likely varies from datagram Version = 4, IP Header Ler 602 to datagram 602. The script is also passed the appropriate datagram 702 template based on the specific software Flag 0x4 DO_NOT_FRAG Fragment Offset = 0x000 register (508 in FIG. 5 or 316 in FIG. 3). There are different IP Protocol = 0x11 (UDP) scripts for different types of datagrams 702 (e.g., UDP or IP Address of Source = 128 TCP). Also, the script would most likely make a copy of the IP Address of Destination = datagram 702 template (not shown here), so that multiple datagrams 602 for the same user could be simultaneously in Source Port = 0x0007 (ech Destination Port = 0x3018 transit.

Paper 2 (036 Petition) at 45, 56, 65-66; Paper 1 (072 Petition) at 41-42, 53; Ex. 1005 (Erickson) at 7:65-8:9, Fig. 7; Ex. 1003.096-.097, .111 (072 Horst Decl.); Ex. 1003.093-.094, .104 -.105 (036 Horst Decl.).

intel)

Erickson: "Pre-negotiated" header template includes "almost everything"



In the present application, the access privileges given to the user processes are very narrow. Each user process has basically pre-negotiated almost everything about the datagram 602, except the actual user data 610. This means most of the fields in the three header areas 604, 606, and 608 are predetermined.

In this example, the user process and the device driver has pre-negotiated the following fields from FIG. 6: (1) Ethernet Header 604 (Target Ethernet Address, Source Ethernet Address, and Protocol Type); (2) IP Header 606 (Version, IP header Length, Service Type, Flag, Fragment Offset, Time__ to_Live, IP Protocol, IP Address of Source, and IP Address of Destination); and (3) UDP Header 608 (Source Port and Destination Port). Only the shaded fields in FIG. 6, and the user data 610, need to be changed on a per-datagram basis.

Paper 2 (036 Petition) at 42, 56-57; Paper 1 (072 Petition) at 40, 53, 63, Ex. 1003.093, -.095 (036 Horst Decl.); Ex. 1003.095 -.096, .112, (072 Horst Decl.); Ex. 1005 (Erickson) at 6:57-7:4, Figure 6.

Erickson: Adapter uses scripts for multiple protocols including TCP/IP

ABSTRACT

A method of controlling an input/output (I/O) device con-

nected to a computer to facilitate fast I/O data transfers. An

address space for the I/O device is created in the virtual

memory of the computer, wherein the address space com-

prises virtual registers that are used to directly control the I/O device. In essence, control registers and/or memory o

the I/O device are mapped into the virtual address space, and

the virtual address space is backed by control registers

and/or memory on the I/O device. Thereafter, the I/O device

detects writes to the address space. As a result, a pre-defined

sequence of actions can be triggered in the I/O device by

programming specified values into the data written into the

19 Claims, 7 Drawing Sheet

mapped virtual address space.

5,768,618

Jun. 16, 1998

[11] Patent Number:

[45] Date of Patent:

United States Patent [19] Erickson et al.

[54] METHOD FOR PERFORMING SEQUENCE OF ACTIONS IN DEVICE CONNECTED TO COMPUTER IN RESPONSE TO SPECIFIED VALUES BEING WRITTEN INTO SNOOPED SUB PORTIONS OF ADDRESS SPACE [75] Inventors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of San New 21 of Oall Diego, all of Calif. [73] Assignce: NCR Corporation. Dayton, Ohio [21] Appl. No.: 577,678

473

[57]

SINGLE PAGE SINULE PAGE

[22] Filed: Dec. 21, 1995 [51] Int. CL⁶ _ G06F 15/02 [52] U.S. CL ...

395/821, 823 [58] Field of Search 395/829, 832, 846, 882, 284, 309, 50

[56] References Cited U.S. PATENT DOCUMENTS

4,589,063 5/1986 Shah et al....... 4,777,589 10/1988 Boetmer et al..... 5,016,161 5/1991 Van Loo et al.... 5,127,088 6/1991 Van Loo et al... 5,127,088 6/1992 Rosenthal et al... 5,220,587 1/1994 Shimodains et al... 5,420,987 5/1998 Reid et al... 395/828 395/823 395/678 395/674 711/202 395/880 395/830 395/823 395/185.01 5.671.442 9/1997 Forney et al. 395/834

FOREIGN PATENT DOCUMENTS 551148 7/1993 European Pat. Off. . OTHER PUBLICATIONS

"The Performance of Message-Passing Using Restricted Virtual Memory Remapping". by Shin-Yuan Tzou and David P. Anderson. in Software-Practice & Experience, vol. 21(3), 251-267 (Mar. 1991).

be performed based upon the protocol type. Each type of protocol will have its own script. Types of protocols include, but are not limited to, TCP/IP, UDP/IP, BYNET lightweight datagrams, deliberate shared memory, active message handler, SCSI, and File Channel

Paper 2 (036 Petition) at 43, 46, 58; Paper 41 (036 Reply) at 2; Paper 1 (072 Petition) at 42, 44, 47, 53; Paper 46 (072 Reply) at 2; Ex. 1003.095, .107, .120 (036 Horst Decl.); Ex. 1003.093, .096, .101 (072 Horst Decl.); Ex. 1005 (Erickson) at 5:41-51.

intel)

Erickson: Identifies Tanenbaum as a reference for TCP

nicate with each other. A discussion of the form and structure of TCP sockets and packets, which are well-known within the art, may be found in many references. including *Computer Networks* by Andrew S. Tanenbaum, Prentice-Hall, New Jersey, 1981, pp. 326–327, 373–377, which is herein incorporated by reference.

> Paper 2 (036 Petition) at 46; Paper 1 (072 Petition) at 34; Ex. 1005 (Erickson) at 4:37-44.

A POSA following Erickson's suggestion would consult the then-current (1996) edition of Tanenbaum to implement Erickson's TCP script

Paper 2 (036 Petition) at 46; Paper 41 (036 Reply) at 5; Paper 1 (072 Petition) at 35; Paper 46 (072 Reply) at 5; Ex. 1003.077 (036 Horst Decl.) ¶ 139; Ex. 1003.079 (072 Horst Decl.) ¶ 138.

18

PO's expert taught Tanenbaum96 before alleged priority date Oct. 1997



Paper 41 (036 Reply) at 10; Paper 46 (072 Reply) at 10; Paper 60 (036 Opp. to Motion to Exclude) at 5-6; Paper 64 (072 Opp. to Motion to Exclude) at 5-6; Ex. 1234 (Almeroth Dep., Ex. 21); Ex. 1225.219 (Almeroth Depo.) at 474:21-475:2.

PO patents describe Tanenbaum96 as a college-level textbook



The above description of layered protocol processing is simplified, as college-level textbooks devoted primarily to this subject are available, such as Computer Networks, Third Edition (1996) by Andrew S. Tanenbaum, which is incor-

> Paper 60 (036 Opp. to Mot. to Exclude) at 13; Paper 64 (072 Opp. to Mot. to Exclude) at 13; Ex. 1001 (036 Patent) at 4:47-50; Ex. 1001 (072 Patent) at 4:57-60; See also Paper 2 (036 Petition) at 34; Paper 41 (036 Reply) at 10; Paper 1 (072 Petition) at 28; Paper 46 (072 Reply) at 10.

> > **intel** 20

Tanenbaum96: "Fast path" processing using a prototype header

THIRD EDITI

ANDREW S. TANENBAL

TO THE NFORMATIC SUPER

HIGHWAY

traps into the kernel to do the SEND. The first thing the transport entity does is make a test to see if this is the normal case: the state is *ESTABLISHED*, neither side is trying to close the connection, a regular (i.e., not an out-of-band) full TPDU is being sent, and there is enough window space available at the receiver. If all conditions are met, no further tests are needed and the <u>fast path</u> through the sending transport entity can be taken.

In the normal case, the headers of consecutive data TPDUs are almost the same. To take advantage of this fact, a prototype header is stored within the transport entity. At the start of the fast path, it is copied as fast as possible to a

Paper 2 (036 Petition) at 35, 47-49; Ex.1003.059, .079-.083 (036 Horst Decl.); Paper 1 (072 Petition) at 28-29,35-38; Ex. 1003.061,.079-.084 (072 Horst Decl.); Ex. 1006.583 (Tanenbaum96).

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

> Paper 2 (036 Petition) at 37, 47-49; Ex.1003.062, -079-.083 (036 Horst Decl.); Paper 1 (072 Petition) at 30,35-38; Ex. 1003.064,-.079-.084 (072 Horst Decl.); Ex. 1006.585 (Tanenbaum96).

Tanenbaum96: Protocol processing is "straightforward" for the "normal case"

6.6.4. Fast TPDU Processing

Provide the second seco

THIRD EDITION

ANDREW S. TANENBAUM

The moral of the story above is that the main obstacle to fast networking is protocol software. In this section we will look at some ways to speed up this software. For more information, see (Clark et al., 1989; Edwards and Muir, 1995; and Chandranmenon and Varghese, 1995).

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.

Paper 2 (036 Petition) at 15, 49, 58, 62; Paper 41 (036 Reply) at 7-8; Paper 1 (072 Petition) at 14-15, 28-29, 37-39; Paper (072 Reply) at 7-8; Ex. 1003.033, .082, .096, .100 (036 Horst Decl.); Ex. 1003.034, .084 (072 Horst Decl.); Ex. 1006.583 (Tanenbaum96).

Tanenbaum96: Transport entity may reside on network interface

THIRD EDITION

NFORMATIC SUPER HIGHWAY by the network layer. The hardware and/or software within the transport layer that does the work is called the **transport entity**. The transport entity can be in the operating system kernel, in a separate user process, in a library package bound into network applications, or on the network interface card. In some cases, the



In general, the transport entity may be part of the host's operating system or it may be a package of library routines running within the user's address space. It may also be contained on a coprocessor chip or network board plugged into the host's backplane. For simplicity, our example has been programmed as though it

Paper 41 (036 Reply) at 6; Paper 46 (072 Reply) at 6; Ex. 1006.530 (Tanenbaum96).

Fast path transmit reuses the prototype header



THIRD EDITION

As an example of how this principle works in practice, let us consider TCP/IP. Fig. 6-50(a) shows the TCP header. The fields that are the same between consecutive TPDUs on a one-way flow are shaded. All the sending transport entity has to do is copy the five words from the prototype header into the output buffer, fill in the next sequence number (by copying it from a word in memory), compute the checksum, and increment the sequence number in memory. It can then hand the header and data to a special IP procedure for sending a regular, maximum TPDU. IP then copies its five-word prototype header [see Fig. 6-50(b)] into the buffer, fills in the *Identification* field, and computes its checksum. The packet is now ready for transmission.

Source port Destination port	VER. IHL TOS Total length		
Sequence number	Identification Fragment offset		
Acknowledgement number	TTL Protocol Header checksum		
Len Unused Window size	Source address		
Checksum Urgent pointer	Destination address		
(a)	(b)		

Fig. 6-50. (a) TCP header. (b) IP header. In both cases, the shaded fields are taken from the prototype without change.

Paper 2 (036 Petition) at 36, 47-49; Paper 1 (072 Petition) at 29, 35-37; Ex. 1003.061, .077-082 (036 Horst Decl.); Ex. 1003.063, .080-.085 (072 Horst Decl.); Ex. 1006.584 (Tanenbaum96).

intel

Tanenbaum96 teaches how to modify Erickson's template header for TCP



intel

Fast path receive updates a connection record and copies data to user memory

COMPUTER NETWORKS

ANDREW S. TANENBAUM



The TPDU is then checked to see if it is a normal one: the state is *ESTAB*-*LISHED*, 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.

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

> Paper 2 (036 Petition) at 37-39, 48-49, 54-63; Paper 1 (072 Petition) at 30-31, 35-37, 70-71; Ex. 1003.060, .078-082 (036 Horst Decl.); Ex. 1003.064, .080-.085 (072 Horst Decl.); Ex. 1006.585 (Tanenbaum96).

The connection record stores TCP state information

COMPUTER NETWORK

THIRD EDITION

ANDREW S. TANENBAUM

heavy solid line) then later the path of a server (the heavy dashed line). When an application on the client machine issues a CONNECT request, the local TCP entity creates a connection record, marks it as being in the *SYN SENT* state, and sends a *SYN* segment. Note that many connections may be open (or being opened) at the same time on behalf of multiple applications, so the state is per connection and recorded in the connection record. When the *SYN+ACK* arrives, TCP sends the



Paper 2 (036 Petition) at 38, 48-49, 54-63; Paper 1 (072 Petition) at 30, 35-37; Ex. 1003.065, .078-082, .091-.101 (036 Horst Decl.); Ex. 1003.065, .080-.085 (072 Horst Decl.); Ex. 1006.549 (Tanenbaum96).

e) 27

The connection record is looked up using the IP addresses and TCP ports

COMPUTER NETWORKS

ANDREW S. TANENBAUM

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.



Paper 2 (036 Petition) at 37-38, 47-49, 54-63; Paper 1 (072 Petition) at 30, 35-37; Ex. 1003.064-.065, .078-082, .091-.101 (036 Horst Decl.); Ex. 1003.064-.065, .080-.085 (072 Horst Decl.); Ex. 1006.585 (Tanenbaum96).

28

Tanenbaum96 teaches how to modify Erickson's endpoint table for TCP

Ur	nited	Sta	tes Patent	[19]	[11]	Patent Number:	5,768,618
Eric	ekson e	t al.			[45]	Date of Patent:	Jun. 16, 1998
[54]	METHO OF AC COMP VALUE SUB PC	OD FOI TIONS I UTER I S BEIN ORTION	R PERFORMING SI IN DEVICE CONNI N RESPONSE TO S G WRITTEN INTO IS OF ADDRESS SI	EQUENCE ECTED TO PECIFIED SNOOPED ACE	"The DA and Shin 1988. Co Californi	ASH Local Kernal Structure -Yuan Tzou, Report No. U omputer Science Division ia. Berkeley 94720.	" by David P. Anderson "B/CSD 88/463, Nov. 7. (EECS), University of
[75]	Inventor	s: Gene Hun Mull Dieg	e R. Erickson; Doug dley, both of Poway; ler; Curtis H. Stehle o, all of Calif.	as E. P. Keith y. both of San	"A Users municati National Box 200 (Aug. 19	s' Guide to PICL—A Porta on Library" By G.A. Ge Laboratory, Mathematical 99. Bidg, 9207–A. Oak F 1990).	ible Instrumented Com- ist et. al., Oak Ridge Sciences Section. P.O. tidge, TN 37831-8083
[21]	Appl. N	o.: 577,0	78	s, Onio	"Archite Stunkel. New Yor	cture and Implementation of et. al., IBM Research Divis (Sep. 22, 1993).	of Vulcan" By Craig B. sion, Yorktown Heights,
[51] [52] [58]	Int. CL ⁴ U.S. Cl. Field of	Search 395	/829, 832, 846, 882,		"MPI-F: by Hube Research	An MPI Prototype Impler ertus Franke et. al., pub. Center, Yorktown Heights	aentation on IBM SP1" by IBM. T.J. Watson 5, New York 10598.
[56]		Re U.S. PAI	ferences Cited		Primary Attorney, Welter &	Examiner—Moustafa M. M. Agent, or Firm—Merchan & Schmidt	feky It, Gould, Smith, Edell,
4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	589.063 (777.589) (016.161) (016.166) (127.098) (280.587) (240.987) (548.778) (553.244) (542.481) (571.442) FOI 551148 Performal P Ander 1 P. Ander	5/1986 10/1988 5/1991 6/1992 1/1994 5/1995 8/1996 9/1996 6/1997 8/1996 6/1997 7/1993 OTHEI ance of ty Remaison, in Ston,	Shab et al. Bocturer et al. Wan Loo et al. Nonando et al. Romandi et al.	95623 956723 956678 956678 956678 956820 956820 956820 955820 955820 3957823 3957834 3957834 3957834 TTS	[57] A metho nected to address a memory prises via I/O devia the I/O d the virtu and/or m detects w sequence programm mapped of	ARSTRACT d of controlling an inputive a comparter to facilitate fa pace for the L/O device i of the computer, wherein rutal registers that are uses evice are mapped into the v ivite so the address space. A of actions can be triggers and address space is back and address space is back and address space is back and address space. I address space.	that (JO) device con- utput (JO) device con- tor Control and transfers. An screated in the virtual the address space con- to directly control fragienes and by control registers of by control registers and the control registers of the control regi
			THE USER PROCESSES		LINEMORY ADDATES E PAGE E PAGE 238		



cesses. Each entry within the endpoint table 514 points to various protocol data 518 in the memory 512 in order to accommodate multiple communication protocols, as well as previously defined protocol scripts 516 in the memory 512, which indicate how data or information is to be transferred from the memory 512 of the I/O device adapter to the portions of main memory 502 associated with a user process.

Paper 2 (036 Petition) at 41-42, 47-49; Paper 1 (072 Petition) at 34-37; Ex. 1003.067-.068, .079-.082 (036 Horst Decl.); Ex. 1003.069-.070, .081-.084 (072 Horst Decl.); Ex. 1005 (Erickson) at 5:53-67, Fig. 5.

intel

Connection record in Tanenbaum96 corresponds to endpoint data in Erickson

Tanenbaum96

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

> Paper 2 (036 Petition) at 37-38, 47-49, 54-63; Paper 1 (072 Petition) at 30, 35-37; Ex. 1003.064-.065, .078-082, .091-.101 (036 Horst Decl.); Ex. 1003.064-.065, .080-.085 (072 Horst Decl.); Ex. 1006.585 (Tanenbaum96).

Erickson

cesses. Each entry within the endpoint table 514 points to various protocol data 518 in the memory 512 in order to accommodate multiple communication protocols, as well as previously defined protocol scripts 516 in the memory 512, which indicate how data or information is to be transferred from the memory 512 of the I/O device adapter to the portions of main memory 502 associated with a user process.

TCP and UDP are alternative protocols for the TCP/IP protocol suite

THIRD EDITIO COMPUTER NET ANDREW S. TANENBAUM	N VORKS	
*		Layer (OSI names)
	TELNET FTP SMTP DNS	Application
Protocols	TCP	Transport
	IP	Network
WELCOME	ARPANET SATNET Packet LAN	Physical + data link
TO THE INFORMATION BUPER HIGHWAY	Fig. 1-19. Protocols and networks in the TCP/IP model initially.	
	Paper 2 (036 Petition) at 21; F Paper 1 (072 Petition) at 39; F	Paper 41 (036 Reply) at 2; Paper 46 (072 Reply) at 2;

Paper 1 (072 Petition) at 39; Paper 46 (072 Reply) at 2; Ex. 1003.060 (072 Horst Decl.); Ex. 1003.057 (036 Horst Decl.); Ex. 1006.055 (Tanenbaum96).

intel

TCP and UDP: "Two main protocols" for IP

ANDREW S. TANENBAUM

THIRD EDITION



6.4. THE INTERNET TRANSPORT PROTOCOLS (TCP AND UDP)

The Internet has two main protocols in the transport layer, a connectionoriented protocol and a connectionless one. In the following sections we will study both of them. The connection-oriented protocol is TCP. The connectionless protocol is UDP. Because UDP is basically just IP with a short header added, we will focus on TCP.



Paper 2 (036 Petition) at 21; Paper 41 (036 Reply) at 3; Paper 1 (072 Petition) at 18-19; Paper 46 (072 Reply) at 3; Ex. 1003.057 (036 Horst Decl.); Ex. 1003.060 (072 Horst Decl.); Ex. 1006.539 (Tanenbaum96).

32

036 Patent: Disputes

- 1. A POSA would have been motivated to combine Tanenbaum96 with Erickson
 - a) A POSA would have naturally looked to Tanenbaum96 when implementing Erickson's TCP functionality
 - b) Tanenbaum96 does not teach away from the invention
 - c) A POSA would have a reasonable expectation of success using Tanenbaum96 to implement Erickson's TCP functionality
 - d) Dr. Horst's 2001 Article shows that "conventional wisdom" was to offload TCP

Tanenbaum96 does not teach away from invention

PO relies on following passage:

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.

Paper 30 (036 Response) at 24-25; Paper 34 (072 Response) at 36-37.

Tanenbaum96 does not teach away from invention

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.

Paper 42 (036 Reply ISO Motion to Amend) at 6-7; Paper 46 (072 Reply) at 6-7; Ex. 1006.588-.589 (Tanenbaum96).

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

6.6.4. Fast TPDU Processing

THIRD EDITION

ANDREW S. TANENBAU

dia alla

TO THE

SUPER

The moral of the story above is that the main obstacle to fast networking is protocol software. In this section we will look at some ways to speed up this software. For more information, see (Clark et al., 1989; Edwards and Muir, 1995; and Chandranmenon and Varghese, 1995).

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.

Paper 2 (036 Petition) at 15, 49, 58, 62; Paper 41 (036 Reply) at 7-8; Paper 1 (072 Petition) at 14-15, 28-29, 37-39; Paper (072 Reply) at 7-8; Ex.1003.033, .082, .096, .100 (036 Horst Decl.); Ex. 1003.034, .084 (072 Horst Decl.); Ex. 1006.583 (Tanenbaum96).

36
PO mischaracterizes the base reference in the combination

PO argues:

Moreover, Petitioner provides no explanation as to how, or indeed, why a

POSITA would have modified *Tanenbaum* in such a way. Petitioner does not

Paper 30 (036 Response) at 26; Paper 34 (072 Response) at 38.

<u>But</u> Petitioner is relying on modification of Erickson:

As set forth below, Erickson in view of Tanenbaum96 renders obvious

claims 1-7 of the 036 Patent. Erickson discloses the large majority of the

limitations through its description of fast path protocol processing of UDP/IP by an

I/O device adapter. Tanenbaum96 discloses corresponding details for TCP/IP. In

Paper 2 (036 Petition) at 50; Paper 41 (036 Reply) at 6; see also Paper 1 (072 Petition) at 38; Paper 46 (072 Reply) at 6.

Tanenbaum96 also teaches offloading transport layer to interface card

THIRD EDITION

SUPER

by the network layer. The hardware and/or software within the transport layer that does the work is called the **transport entity**. The transport entity can be in the operating system kernel, in a separate user process, in a library package bound into network applications, or on the network interface card. In some cases, the



In general, the transport entity may be part of the host's operating system or it may be a package of library routines running within the user's address space. It may also be contained on a coprocessor chip or network board plugged into the host's backplane. For simplicity, our example has been programmed as though it

Paper 41 (036 Reply) at 6; Paper 46 (072 Reply) at 6; Ex. 1006.530 (Tanenbaum96).

Dr. Horst: Tanenbaum96 does not teach away from the combination

Q. But you wouldn't consider TCP to be an exceedingly simple protocol, would you?

A. The fast path of TCP that's only transferring data is not that complicated. Even the full TCP, there are plenty of examples of people that have solved the problems Tanenbaum is talking about and have done all kinds of different levels of off-loading as I described in my introductory section of the report.

intel

Erickson's benefits apply equally to TCP

United States Patent [19]	[11] Patent Number:
Erickson et al.	[45] Date of Patent:
[54] METHOD FOR PERFORMING SEQUENCI OF ACTIONS IN DEVICE CONNECTED TO COMPUTER IN RESPONSE TO SPECIFIE VALUES BEING WRITTEN INTO SNOOP SUB PORTIONS OF ADDRESS SPACE	 "The DASH Local Kernal Structure" to and Shin-Yuan Tzou, Report No. UCB. 1988. Computer Science Division (E California, Berkeley 94720.
 [75] Inventors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of 3 Diego, all of Calif. [73] Assigner: NCR Conversion Dation Obio. 	"A Users' Guide to PICL—A Portable munication Library" By G.A. Geist National Laboratory, Mathematical So Box 2009, Bldg, 9207–A. Oak Ridq (Aug. 1990).
 [21] Appl. No.: 577,678 [22] Filed: Dec. 21, 1995 	"Architecture and Implementation of Stunkel, et. al., IBM Research Division New York (Sep. 22, 1993).
[51] Int. CL ⁶ G06F 1: [52] U.S. Cl. 395/ [58] Field of Search 395/829, 832, 846, 882, 284, 309, :	5/02 "MPI-F: An MPI Prototype Implement by Hubertus Franke et. al., pub. by Research Center, Yorktown Heights, N 473
[56] References Cited U.S. PATENT DOCUMENTS	Primary Examiner-Moustafa M. Mel Attorney, Agent, or Firm-Merchant, Welter & Schmidt
4.58000 57046 The et al	NEL [57] ABSTRACT NEL A method of controlling an inputtoom NEL A control of control registers NEL D device. In escence. control register NEL D device. In escence.



Thus, it will be recognized that the present invention increases the efficiency of I/O operations in the following ways:

- 1. Writing information to and from a user address space without intermediate memory-to-memory copies.
- 2. Accessing an I/O device simultaneously from multiple user processes in a single node.
- 3. Eliminating calls to the operating system and the associated context switches on a per I/O basis.
- 4. Maintaining system security for the I/O device by using the operating system to initialize the virtual memory address space of the user process.
- 5. Accessing the I/O device under the full control of the resource allocation policies and permissions granted by the operating system.
- 6. Working with a plurality of well-known standard operating systems including, but not limited to, UNIX, OS/2, Microsoft Windows, Microsoft Windows NT, or Novell Netware.
- 7. Providing low-latency high-performance control of I/O devices.

40

036 Patent: Disputes

- 1. A POSA would have been motivated to combine Tanenbaum96 with Erickson
 - a) A POSA would have naturally looked to Tanenbaum96 when implementing Erickson's TCP functionality
 - b) Tanenbaum96 does not teach away from the invention
 - c) <u>A POSA would have a reasonable expectation of success</u> <u>using Tanenbaum96 to implement Erickson's TCP</u> <u>functionality</u>
 - d) Dr. Horst's 2001 Article shows that "conventional wisdom" was to offload TCP

Tanenbaum96 identified freely available TCP/IP source code: Berkeley (BSD) UNIX

Tanenbaum96 (1996)

In contrast, one of the first implementations of TCP/IP was part of Berkeley UNIX[®] and was <u>quite good (not to mention, free)</u>. People began using it quickly, which led to a large user community, which led to improvements, which led to an even larger community. Here the spiral was upward instead of downward.



THIRD EDITION

ANDREW S. TANENBAUM

Paper 2 (036 Petition) at 15; Paper 41 (036 Reply) at 10; Paper 1 (072 Petition) at 18; Paper 46 (072 Reply) at 10; Ex. 1003.013, .020-.021 (036 Horst Decl.) ¶¶ 26, 34; Ex. 1223.011-.014 (036 Horst Reply Decl.) ¶¶ 26-29. Ex. 1003.014 (072 Horst Decl.) ¶ 26 ; Ex. 1223.011-.014 (072 Horst Reply Decl.) ¶¶ 26-29; Ex. 1006.061 (Tanenbaum96).

intel

Tanenbaum96: Fast path/header prediction is widely used

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



THIRD EDITION

ANDREW S. TANENBAUM

Paper 41 (036 Reply) at 10; Paper 46 (072 Reply) at 10; Ex. 1003.039 (036 Horst Decl.) ¶ 70; Ex. 1003.040 (072 Horst Decl.) ¶ 70; Ex. 1006.585 (Tanenbaum96).



Berkeley TCP included fast-path header prediction code

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

67. Code to implement the header prediction algorithm was incorporated

in the BSD 4.4-Lite distribution.

Case IPR. No. **Unassigned** U.S. Patent No. 7,237,036 Title: FAST-PATH APPARATUS FOR RECEIVING DATA CORRESPONDING A TCP CONNECTION

> Declaration of Robert Horst, Ph.D. in Support of Petition for *Inter Partes* Review of U.S. Patent No. 7,237,036

Paper 41 (036 Reply) at 10; Paper 46 (072 Reply) at 10; Ex. 1003.037-.038 (036 Horst Decl.) ¶ 67; Ex. 1003.038-.039 (072 Horst Decl.) ¶ 67.

Other college textbooks documented Berkeley TCP/IP



This book describes and presents the source code for the common reference implementation of TCP/IP: the implementation from the Computer Systems Research Group (CSRG) at the University of California at Berkeley. Historically this has been distributed with the 4.x BSD system (Berkeley Software Distribution). This implementation was first released in 1982 and has survived many significant changes, much fine tuning, and numerous ports to other Unix and non-Unix systems. This is not a toy implementation, but the foundation for TCP/IP implementations that are run daily on hundreds of thousands of systems worldwide. This implementation also provides router functionality, letting us show the differences between a host implementation of TCP/IP and a router. We describe the implementation and present the entire source code for the kernel

implementation of TCP/IP, approximately 15,000 lines of C code. The version of the

Paper 2 (036 Petition) at 15; Paper 1 (072 Petition) at 18;

Paper 41 (036 Reply) at 10); Paper 46 (072 Reply) at 10;

Th

Gag

Ex. 1003.013 (036 Horst Decl.) ¶ 26; Ex. 1003.014-.015 (072 Horst Decl.) ¶ 26;

Ex. 1223.011-.014 (036 Horst Reply Decl.) ¶¶ 26-29; Ex. 1223.011-.014 (072 Horst Reply Decl.) ¶¶ 26-29; Ex.1013.023 (Stevens2).

Stevens2 documented the BSD header prediction code

Stevens Vol. 2 (1995)

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

68. The 1995 book (Stevens2) walks through the Jacobson BSD header

prediction code including the conditions for selecting the fast or slow path. In order

Case IPR. No. **Unassigned** U.S. Patent No. 7,237,036 Title: FAST-PATH APPARATUS FOR RECEIVING DATA CORRESPONDING A TCP CONNECTION

> Declaration of Robert Horst, Ph.D. in Support of Petition for *Inter Partes* Review of U.S. Patent No. 7,237,036

Paper 41 (036 Reply) at 10; Paper 46 (072 Reply) at 10; Ex.1003.038-.039 (036 Horst Decl.) ¶ 68; Ex. 1003.039-.040 (072 Horst Decl.) ¶ 68.

46

intel

User data does not need to span page boundaries to support TCP

• PO claims:

column 8. (Ex. 2026, ¶ 94.) Erickson itself acknowledges that its script "would

need to be enhanced if the user data were allowed to span page boundaries" (Ex.

1005 at 8:22-24)—something Petitioner's own expert admits is not described by

Paper 30 (036 Response) at 29; Paper 34 (072 Response) at 41.

• But data does not need to span page boundaries for TCP:

embodiments in the specification. However, even if Erickson was limited to a single page, TCP segments are often smaller than a page. For example, in Ethernet which is the most common media, TCP segments are typically about 1500 bytes, which is smaller than a typical page size of 4K bytes. Thus, a POSA could have implemented a TCP embodiment without making changes to the Erickson adapter's ability to cross page boundaries.

Paper 41 (036 Reply) at 12; Paper 46 (072 Reply) at 11-12); Ex. 1223.019-.020 (036 Horst Reply Decl.) ¶ 40; Ex. 1223.019-.020 (072 Horst Reply Decl.) ¶ 40.



Spanning page boundaries would have been straightforward design choice

• Would merely require multiple calls to "vtophys" function disclosed in Erickson

Header 706 portion of the datagram 702. The vtophys() function performs a translation of the user-provided virtual address into a physical address usable by the adapter. In all

Paper 41 (036 Reply) at 12; Paper 46 (072 Reply) at 12; Ex. 1005 (Erickson) at 8:14-16.

Erickson contemplates spanning page boundaries

be fixed. The udpscript procedure would need to be enhanced if the user data were allowed to span page boundaries. The udpchecksum() procedure generates a checksum



intel

Need design details before you can predict speed

Patent Owner argues:

During his deposition, Dr. Horst also testified that it is impossible to

determine whether executing TCP code on Erickson's adapter would be slower or

faster than the traditional slow path:

<u>But</u> speed is unpredictable without design parameters:

A. Just looking at the code, you couldn't tell. You also need to have information on the compiler, the processor used, the caches. There's all kinds of things that influence the performance of code.

> Paper 30 (036 Response) at 40; Paper 34 (072 Response) at 52; Ex. 2029 (Horst Dep.) at 81:23-82:9; Paper 41 (036 Reply) at 13; Paper 46 (072 Reply) at 12-13.



Combination does not have to be predictably "faster"

Networks are built around standards that specify links with fixed 44. speeds. Over time, the Ethernet physical layer has evolved from 1 mbit/sec, to 10 mbit/sec, to 100 mbit/sec, to Gbit/sec and 10 Gbit/sec. At a given point in time, offloading the networking protocol does not increase the link speed, which must support the then-current standard, but may reduce the processing load on the main CPU and transfer some of the load to the network adapter, freeing the main processor to perform other work. In multiprocessor systems, moving the load between processors is called "load balancing," and the same concept can be applied to networking if the network adapter has sufficient intelligence to take on part of the load. The goal of reducing host processing load has been one of the reasons for much of the prior work on protocol offloads. Guerrero and others have

Definition of POSA does not matter for obviousness determination

Petitioner:

and/or networking protocols. PO's Response at 23. While I disagree with this proposed level of ordinary skill, my opinions in this declaration would remain the same even if Patent Owner's opinion concerning the level of ordinary skill in the art were applied.

Ex. 1210.005-.006 (036 Horst Decl. ISO Opp. to Motion to Amend) ¶ 9; Ex. 1210.005 (072 Horst Decl. ISO Opp. to Motion to Amend) ¶ 9; Ex. 1223.009 (036 Horst Reply Decl.) ¶ 21; Ex. 1223.009 (072 Horst Reply Decl.) ¶ 21.

Patent Owner:

computer networking and/or networking protocols. (Ex. 2026 at ¶ 33.) Any differences between this and Petitioners' proposed level of ordinary skill would have no bearing on the analysis presented in this Response. The cited references

Paper 30 (036 Response) at 22; Paper 34 (072 Response) at 23.

036 Patent: Disputes

- 1. A POSA would have been motivated to combine Tanenbaum96 with Erickson
 - a) A POSA would have naturally looked to Tanenbaum96 when implementing Erickson's TCP functionality
 - b) Tanenbaum96 does not teach away from the invention
 - c) A POSA would have a reasonable expectation of success using Tanenbaum96 to implement Erickson's TCP functionality
 - d) <u>Dr. Horst's 2001 Article shows that "conventional wisdom"</u> was to offload TCP

e) 52

Industry actively working on offloading TCP/IP

IP Storage and the CPU Consumption Myth

Robert Horst 3ware, Inc. 701 E. Middlefield Rd. Mountain View, CA 94043

Abstract

This paper addresses a key issue a attaching storage devices directly to h perceived need for hardware acceleratin networking stack. While many implic acceleration is required, the evidence conclusion is not well founded. In the accelerators have had mixed success, economic justification for hardware acc given the low cost of 0 sot CPU cycles. many applications is dominated by o transfer rate, and hardware protocol a little effect on the 1/O performance in the Application benchmarks were run on subsystem to measure performance and the results show that good performance those those those the source of the cost of the the subsystem to measure performance and the results show that good performance.

1. Introduction

The growing popularity of gigab prompted increasing interest in usin networks to attach storage devices to Ethernet Storage Area Networks (significant advantages in cost and m compared with Fibre Chamel SANs, products are already on the market standardize the protocols is progressing working group of the IETF [1]. Networks customized to storage net

Fiber Channel, were developed larg perception that standard networking protocols are too heavyweight for attaching storage. Conventional wisdom says that IP storage is impractical without special purpose NICs to accelerate the TC/PI protocol stack. This papers shows that the need for hardware acceleration is largely a myth. Several different lines of reasoning show that the future of storage networking will rely heavily on storage devices connected to servers without special purpose hardware accelerators.

Networks customized to storage networking, such as Fiber Channel, were developed largely due to the perception that standard networking protocols are too heavyweight for attaching storage. Conventional wisdom says that IP storage is impractical without special purpose NICs to accelerate the TCP/IP protocol stack. This papers

kills the front-end processor architecture, until the next generation of engineers rediscovers the idea and repeats the cycle.

Some may argue that the problem was that the acclerators should have been optimized hardware instead of embedded programmable processors. Unfortunately, every protocol worthy of acceleration continues to evolve, and it is difficult to stay ahead of the moving target. The new protocols proposed for IP storage, ISCSI and iFCP, are far from stable, and even after the standards have been formally approved, there will likely be a long series of enhancements and bug fixes. It seems extremely Paper 41 (036 Reply) at 14; Paper 46 (072 Reply) at 13-14; Ex. 2300 (IP Storage and the CPU Consumption Myth) at 1.

Dr. Horst's article was focused on networked storage, not other markets

IP Storage and the CPU Consumption Myth

Robert Horst 3ware, Inc. 701 E. Middlefield Rd. Mountain View, CA 94043

Abstract

This paper addresses a key issue a attaching storage devices directly to h perceived need for hardware acceleration acceleration is required, the evidence conclusion is not well founded. In the accelerators have had mised success, economic justification for hardware auc given the low cost of 0 sor CPU cycles. many applications is dominated by a furstifer rate, and hardware protocol a little effect on the 1/0 performance in the Application benchmarks were run on subsystem to measure performance and on Email, database, file serving, and bac The results show that good performance

1. Introduction

The growing popularity of gizab prompted increasing interest in usin networks to attach storage devices to Ethernet Storage Area Networks of significant advantages in cost and rn compared with Fibre Channel SANs. products are already on the market standardize the protocols is progressing working group of the IETF [1].

Networks customized to storage net fiber Channel, were developed larg perception that standard networking p heavyweight for attaching storage. Conv says that IP storage is impractical withou NICs to accelerate the TCP/IP protocol is shows that the need for hardware acceler myth. Several different lines of reasoni future of storage networking will rely b devices connected to servers without hardware accelerators. The growing popularity of gigabit Ethernet has prompted increasing interest in using standard IP networks to attach storage devices to servers. These Ethernet Storage Area Networks (E-SANs), have significant advantages in cost and management ease compared with Fibre Channel SANs. Some IP storage products are already on the market, and work to standardize the protocols is progressing in the IP Storage working group of the IETF [1].

Paper 41 (036 Reply) at 14-15; Paper 46 (072 Reply) at 14; Ex. 2300 (IP Storage and the CPU Consumption Myth) at 1.

036 Patent: Disputes

- 2. The combination of Erickson and Tanenbaum96 discloses the limitations of claims 1-7 of the 036 Patent
 - a) <u>The prior art: "said communication processing mechanism</u> <u>containing a second processor" (all claims)</u>
 - b) The prior art discloses "[second processor] running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory" (all claims)
 - c) The prior art discloses "the TCP state information is updated by said second processor" (all claims)
 - d) The prior art discloses a "receive sequencer with directions to classify said packet" on "said communication processing mechanism" (claim 2)
 - e) The prior art discloses a "receive sequencer with directions to generate a summary" on "said communication processing mechanism (claim 3)

"Said communication processing mechanism containing a second processor"

(12)	Unite Boucher	d St et al.	ates Patent	(10) (45)	Paten Date o	t No.: of Pate	US nt:	7,237, Jun.	036 B2 26, 2007		
(54)	FAST-PAI DATA CO CONNEC	'H APP/ RRESP TION	ARATUS FOR RECEIVING ONDING A TCP	(56)	U.3	Refer S. PATEN 12/198	ences Cite T DOCUI 2 Johnson	ed MENTS et al		1	A devic
(75)	Inventors:	Lauren (US); S Jose, C. Franciso Saratog: San Jos Milpitas	ce B. Boucher, Saratoga, CA tephen E. J. Blightman, San A (US): Peter K. Craft, San 20, CA (US): David A. Higgen, a, CA (US): Clive M. Philbrick, e, CA (US): Davyl D. Starr, s, CA (US)	wo	FORE WO/9	(Co EIGN PAT 8/19412 (Co OTHER P	ontinued) ENT DOC 5/199 ontinued) UBLICAT	CUMENT 8 IONS	at m	ole ei	e to a sec
(73)	Assignee:	Alacrit	ech, Inc., San Jose, CA (US)	Internet (which]	pages enti Eugene Fei	tled "Hard	ware Assist	ed Protoco	111		inory and
(*)	Notice:	Subject patent i U.S.C.	to any disclaimer, the term of this s extended or adjusted under 35 154(b) by 672 days.	1998.		(Co	ontinued)	, Infer In	pr	0	cessing la
(21)	Appl. No.:	10/260,	112	Primary Assistan (74) Att	v Examine nt Examin torney, Aş	er—Jeffre er—Jude gent, or F	y Pwu Jean-Gille <i>irm</i> —Mar	s k Lauer; S	th	e	context i
(22)	Filed:	Sep. 27	, 2002	Law G	oup LLP					1 1	. I
(65)	115 2004/0	Prior	Publication Data	(57) A system	m for prot	AB:	STRACT	computer	ac	ld	ress, an I
	Rel	ated IIS	Annlication Data	an intel	ligent net	work inter	face card	(INIC) or atod with	0		í Í D
(63)	Continuatio Mar. 6, 20	on of app 02, now	blication No. 10/092,967, filed on Pat. No. 6,591,302, which is a	puter. T process acceler	The INIC ing for mating data	provides nost large communi	a fast-pat multi-pac	h that avo ket messa e INIC ab	C	01	itrol Prot
		(C	ontinued)	host for	those me	essage pac	kets that a	re chosen	nr	is	sino
(60)	Provisional 27, 1998,	applicat	ion No. 60/098,296, filed on Aug. nal application No. 60/061,809,	for a m INIC to docting	essage is o move da	defined the ata, free o	at allows I of headers,	DMA cont directly	Р	10	, <u>,</u>
(51)	Int. Cl.	a. 14, 15	(2006.01)	INIC as passed	a commi back to th	unication the host for	control blo message j	ock (CCB		a	commur
(52)	G06F 15/1	7	(2006.01) (2006.01) 709/245: 709/236: 709/230:	much f	aster at the	heir specia d embodir	fic tasks the	nan a gen			first pro
(52)	0.5. 01		370/474; 370/396; 370/469	process	ors with	separate	processors	devoted			inter pro
(58)	Field of C 70	lassifica 9/236, 2	tion Search 709/245, 30, 202; 370/474, 230, 396, 469; 707/2-4, 10; 712/19, 52	munica	tion for fo	our fast E	hernet no	des.			nism co
	See applic	ition file	for complete search history.		22	Claims, 8	9 Drawin	g Sheets			tions to
				15	2						• •
				FAST-PATH	soi	URCE/DES	r 🔒				is empl
			159	1661	AP	PLICATIO	4				the first
			150 170 185	164	Т	RANSPORT					the first
			PROCESSOR S	160	<u> </u>	NETWORK	-				tion is
			HARDWARE LOGIC	LOW-PATH		LINK					tion is
			155 171	L157		138		L		_	

1. A device for use with a first apparatus that is connectable to a second apparatus, the first apparatus containing a memory and a first processor operating a stack of protocol processing layers that create a context for communication, the context including a media access control (MAC) layer address, an Internet Protocol (IP) address and Transmission Control Protocol (TCP) state information, the device comprising:

communication processing mechanism connected to the first processor, said communication processing mechanism containing a second processor running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory and the TCP state information is updated by said second processor.

Ex. 1001 (036 Patent), Claims 1-7.

Erickson discloses a second processor



SINGLE PAGE

KRUSER PROC SECURE VEW A script is prepared by the operating system for the I/O device adapter to execute each time the specific user process programs its specific virtual hardware. The user process is given a virtual address in the user process' address space that allows the user process very specific access capabilities to the I/O device adapter.

> Paper 2 (036 Petition) at 65; Paper 41 (036 Reply) at 15; Ex. 1005 (Erickson) at 4:18-23.

Erickson shows second processor on adapter executes scripts

memory on the I/O device adapter. If the I/O device adapter detects access to the physical memory page, a predefined script is then executed by the I/O device adapter in order to "The l and Shi 1988. Califo direct the data as appropriate. "A Use munica Natio

entors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of San Direce all of Colf. Box 2009. Bldg. 9207-A. Oak Ridge. TN 37831-8083 (Aug. 1990). "Architecture and Implementation of Vulcan" By Craig B. Stunkel, et. al., IBM Research Division, Yorktown Heights, New York (Sep. 22, 1993) __ G06F 15/02 "MPI-

Paper 41 (036 Reply) at 16; Ex. 1005 (Erickson) at 5:37-40.

Paper 41 (036 Reply) at 16;

Ex. 1005 (Erickson) at 7:41-44.

device adapter's memory. The user process provides the starting address and the length for the user data in its virtual address space, and then "spanks" a GO register to trigger the I/O device adapter's execution of a predetermined script.

and/or detects writes to the address space. As a result, a pre-defined sequence of actions can be triggered in the I/O device by programming specified values into the data written into the "The Performance of Message-Passing Using Restricted Virtual Memory Remapping", by Shin-Yuan Tzou and David P. Anderson, in Software-Practice & Experience, vol. 21(3), 251-267 (Mar. 1991). mapped virtual address space.

19 Claims, 7 Drawing Sheets



United States Patent [19]

[54] METHOD FOR PERFORMING SEQUENCE OF ACTIONS IN DEVICE CONNECTED TO COMPUTER IN RESPONSE TO SPECIFIED

Diego, all of Calif.

[58] Field of Search ______ 395/821, 823, 395/829, 832, 846, 882, 284, 309, 500,

References Cited

U.S. PATENT DOCUMENTS

FOREIGN PATENT DOCUMENTS

OTHER PUBLICATIONS

551148 7/1993 European Pat. Off.

[73] Assignce: NCR Corporation. Dayton, Ohio

VALUES BEING WRITTEN INTO SNOOPED SUB PORTIONS OF ADDRESS SPACE

395/829

by H

473 Prime

.... 395/828 395/823 395/678 395/674 711/202 [57]

395/185.01 - 395/834 Resear

Attorna Welter

A met nocted address

memory

prises I/O dev

the I/O

Erickson et al.

[21] Appl. No.: 577,678

[22] Filed: Dec. 21, 1995 [51] Int. CL⁶ [52] U.S. Cl.

> exception for the system bus controller. The bus controller then transfers the data to the I/O device adapter and initiates the registers of the I/O device adapter to execute a predetermined script to process the data.

> > Paper 41 (036 Reply) at 16; Ex. 1005 (Erickson) at 8:54-57

036 Patent: Disputes

- 2. The combination of Erickson and Tanenbaum96 discloses the limitations of claims 1-7 of the 036 Patent
 - a) The prior art discloses "said communication processing mechanism containing a second processor" (all claims)
 - b) <u>The prior art discloses "[second processor] running instructions to</u> process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory" (all claims)
 - c) The prior art discloses "the TCP state information is updated by said second processor" (all claims)
 - d) The prior art discloses a "receive sequencer with directions to classify said packet" on "said communication processing mechanism" (claim 2)
 - e) The prior art discloses a "receive sequencer with directions to generate a summary" on "said communication processing mechanism (claim 3)

"[Second processor] running instructions to process a message packet..."

(12)	United States Patent Boucher et al.	(10) Patent No.: US 7,237,0 (45) Date of Patent: Jun. 2	36 B2 6, 2007
(54)	FAST-PATH APPARATUS FOR RECEIVING DATA CORRESPONDING A TCP CONNECTION	(56) References Cited U.S. PATENT DOCUMENTS	1
(75)	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, Saratoga, CA (US); David A. Higgen, Sana Jose, CA (US); Daryl D. Starr, Milpitas, CA (US)	4.366.538 A 12/1982 Johnson et al (Continued) FOREIGN PATENT DOCUMENTS WO WO98/19412 5/1998 (Continued) OTHER PUBLICATIONS	able t
(73)	Assignee: Alacritech, Inc., San Jose, CA (US)	Internet pages entitled "Hardware Assisted Protocol 1 (which Eugene Feinberg is working on), 1 page, print 1998	
(*) 	Notice: Subject to any disclaimer, the term of the patent is extended or adjusted under 3: U.S.C. 154(b) by 672 days.	5 (Continued)	proce
21)	Appl. No.: 10/260,112	Primary Examiner—Jettrey Pwu Assistant Examiner—Jude Jean-Gilles (74) Attorney, Agent, or Firm—Mark Lauer, Si	the co
(22)	Filed: Sep. 27, 2002	Law Group LLP	11
(65)	Prior Publication Data US 2004/0073703 A1 Apr. 15, 2004	(57) ABSTRACT A system for protocol processing in a computer n	addre
	Related U.S. Application Data	an intelligent network interface card (INIC) or co tion processing device (CPD) associated with a	Cont
(63)	Continuation of application No. 10/092,967, filed on Mar. 6, 2002, now Pat. No. 6,591,302, which is	puter. The INIC provides a fast-path that avoid processing for most large multi-packet messag accelerating data communication. The INIC also	Conti
	(Continued)	host for those message packets that are chosen fo ing by host software layers. A communication co	nrisir
(60)	Provisional application No. 60/098,296, filed on Aug 27, 1998, provisional application No. 60/061,809 filed on Oct. 14, 1997	ing by now sortware injense recommunication composition of a message is defined that allows DMA control. INIC to move data, free of headers, directly to destination or source in the host. The context is st	PIISI
51)	Int. Cl.	INIC as a communication control block (CCB) passed back to the host for message processing b	a c
(52)	G06F 15/18 (2006.01) G06F 15/17 (2006.01) U.S. Cl. 709/245; 709/236; 709/236;	The INIC contains specialized hardware circui much faster at their specific tasks than a gener CPU. A preferred embodiment includes a trio o	t
(58)	370/474; 370/396; 370/469 Field of Classification Search	 processors with separate processors devoted to receive and management processing, with full du munication for four fast Ethernet nodes. 	-
	707/2-4, 10; 712/19, 52 See application file for complete search history.	2 22 Claims, 89 Drawing Sheets	1
			1
		FAST.PATH	
	159	168 APPLICATION]
	150 170 185	166 TRANSPORT	1
	PROCESSOR S	160 NETWORK SLOW-PATH DATA LINK	
		·	

1. A device for use with a first apparatus that is connectable to a second apparatus, the first apparatus containing a memory and a first processor operating a stack of protocol processing layers that create a context for communication, the context including a media access control (MAC) layer address, an Internet Protocol (IP) address and Transmission Control Protocol (TCP) state information, the device comprising:

communication processing mechanism connected to the first processor, said communication processing mechanism containing a second processor running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory and the TCP state information is updated by said second processor.

Ex. 1001 (036 Patent), Claim 1.

PO does not address prior art combination petitioner relies on

U.S. Pat. No. 5,768,618 ("Erickson") in view of Tanenbaum96

[1.3] [second processor] running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory and

Accordingly, Erickson in view of Tanenbaum96 disclose *running instructions* (specified by the scripts) *to process a message packet such that the context* (including the script, registers 508 and 504, endpoint table 514, endpoint protocol data 518, pre-negotiated information, and pointer to main memory) *is employed* (to identify the relevant protocol script to run, and further to identify where to write the received data) *to transfer data contained in said packet to the first apparatus memory* (memory of host computer)

> Paper 2 (036 Petition) at 66-68; Paper 41 (036 Reply) at 17; Ex. 1003.109 (036 Horst Decl.).

Erickson's adapter stores protocol information for moving data to the host

United States Patent [19]						Patent Number:	5,	768,618	
Erickson et al.				[45]		Date of Patent: Ju	m.	16, 1998	
[54]	METH OF AC COMP VALUE SUB P	OD FO TIONS UTER I S BEE ORTIO	R PERFORMING SEQUENCE IN DEVICE CONNECTED TO IN RESPONSE TO SPECIFIED NG WRITTEN INTO SNOOPED NS OF ADDRESS SPACE	"The DA and Shin 1988. C Californi	ia ia	SH Local Kernal Structure" by D Yuan Tzou, Report No. UCB/CSI mputer Science Division (EECS 1. Berkeley 94720.	ivid) 88/). U	P. Anderson (463, Nov. 7, (niversity of	
[75] Inventors: Gene R. Erickson; Douglas E. Hundley, both of Poways P. Keith Muller; Curtis H. Stehley, both of San Diego, all of Calif.				"A Users' Guide to PICL—A Portable Instrumented Com- munication Library" By G.A. Geist et. al., Oak Ridge National Laboratory. Mathematical Sciences Section, PO. Box 2009, Bidg. 9207–A. Oak Ridge, TN 37831–8083 (Aue. 1900).					
[21]	Appl. N	io.: 577,	678 21 1995	"Archite Stunkel. New Yor	ct et k	tare and Implementation of Valc t. al., IBM Research Division, Yo (Sep. 22, 1993).	nn" rkto	By Craig B. wn Heights,	
[51] [52] [58]	Int. CL U.S. Cl Field of	6 T Search 39	G06F 15/02 395/829 395/821, 823, 5/829, 832, 846, 882, 284, 309, 500, 473	"MPI-F: by Hube Research		An MPI Prototype Implementation tus Franke et. al., pub. by IB Center, Yorktown Heights, New	en os M. Yord	n IBM SP1" T.J. Watson t 10598.	
[56]		R U.S. PA	eferences Cited	Primary Attorney, Welter &	E	Zxaminer—Moustafa M. Meky Agent, or Firm—Merchant, Gou Schmidt	d. S	mith. Edell,	
*****	589,063 ,777,589 ,016,161 ,016,166 ,127,098 ,280,587 ,420,987	5/1986 10/1988 5/1991 5/1991 6/1992 1/1994 5/1995	Shah et al.	[57] A metho nected to address s memory	d si si o r	ABSTRACT of controlling an input/output () a computer to facilitate fast I/O d vace for the I/O device is creat of the computer, wherein the add	/O) ata t d in ress	device con- ransfers. An the virtual space com-	
5,5,5,	553,244 642,481 671,442 FO 551148	8/1996 9/1996 6/1997 9/1997 REJGN 7/1993	Parayama	prises via I/O devia the I/O da the virtu and/or m detects w		aal registers that are used to dir. . In essence, control registers an vice are mapped into the virtual ad 1 address space is backed by o mory on the I/O device. Thereafth ices to the address space. As a ress	× 4 × 4	ces	
"The Virtua David 21(3).	Perform al Memo P. Ander 251-26	OTHE ance of ry Rem rson. in 3 7 (Mar.	R PUBLICATIONS Message-Passing Using Restricted appeing [*] , by Shin-Yuan Tzou and Software-Practice & Experience, vol. 1991).	sequence program mapped	vi vi	of actions can be triggered in th ing specified values into the data irtual address space. 19 Claims, 7 Drawing Shee		var	
			THE LEAR PROCESSES VIETURE MEMORY INTO VIETURE LEADWRITE PROCESS VIETURE LEADWRITE PROCESSES	ADAPTER		OEVOLOMAR WETLA ISACOV BURZHUL		pre	
			- 254	IPAGE				wh	
			PER JER PROCESS SECURE WEN TO VIETURE INFORMATE	EPHOE			1	fre	

ADAPTER MEMORY ~ 512 MAIN MEMORY SOFTWARE 516 REGISTER ENDPOINT TABLE INDEXED BY PROTOCOL 508 J HARDWARE R / W APPLICATION ID SCRIPTS APPLICATION RO APPLICATION R / W - 514 HARDWARE RO HARDWARE REGISTER ENDPOINT 504 PROTOCOL DATA .510 506 BUFFER PHYSICAL ADDRESS OS DRIVER R / W POOL BUFFER MAP 518 HARDWARE RO - 502

cesses. Each entry within the endpoint table 514 points to various protocol data 518 in the memory 512 in order to accommodate multiple communication protocols, as well as previously defined protocol scripts 516 in the memory 512, which indicate how data or information is to be transferred from the memory 512 of the I/O device adapter to the portions of main memory 502 associated with a user process.

> Paper 2 (036 Petition) at 67; Ex. 1005 (Erickson) at 5:61-67.

> > 62

intel

It would be obvious to use Tanenbaum96's fast-path connection records with Erickson

THIRD EDITION

TTTT

ANDREW

NFORMATIC SUPER HIGHWAY 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,

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

> Paper 2 (036 Petition) at 60-61; Paper 41 (036 Reply) at 17; Ex. 1003.098-.099 (036 Horst Decl.) at A-12 –A-13; Ex. 1006.584-.585 (Tanenbaum96).

036 Patent: Disputes

- 2. The combination of Erickson and Tanenbaum96 discloses the limitations of claims 1-7 of the 036 Patent
 - a) The prior art discloses "said communication processing mechanism containing a second processor" (all claims)
 - b) The prior art discloses "[second processor] running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory" (all claims)
 - c) <u>The prior art discloses "the TCP state information is updated by said</u> <u>second processor" (all claims)</u>
 - d) The prior art discloses a "receive sequencer with directions to classify said packet" on "said communication processing mechanism" (claim 2)
 - e) The prior art discloses a "receive sequencer with directions to generate a summary" on "said communication processing mechanism (claim 3)

"The TCP state information is updated by said second processor"

(12)	United States Patent	(10) Patent No.: US 7,237,	036 B2
-	soucher et al.	(45) Date of Fatent: Jun.	
(54)	FAST-PATH APPARATUS FOR RECEIVING DATA CORRESPONDING A TCP CONNECTION	(56) References Cited U.S. PATENT DOCUMENTS	1 /
(75)	Inventors: Laurence B. Boucher, Saratoga, CA (US); Stephen E. J. Bilghtman, San Jose, CA (US); Peter K, Craft, San Francisco, CA (US); David A. Higgen, Saratoga, CA (US); David A. Higgen, San Jose, CA (US); Davyl D. Starr, Milpins, CA (US)	4,366,538 A 12/1982 Johnson et al (Continued) FOREIGN PATENT DOCUMENT: WO W0.98/19412 5/1998 (Continued) OTHER PUBLICATIONS	able to
(73)	Assignee: Alacritech, Inc., San Jose, CA (US)	Internet pages entitled "Hardware Assisted Protoco (which Eugene Feinberg is working on), 1 page, pri	meme
(*)	Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 672 days.	1998. (Continued)	proces
(21)	Appl. No.: 10/260,112	Assistant Examiner—Jude Jean-Gilles (74) Attorney, Agent, or Firm—Mark Lauer, S	the co
(22)	Filed: Sep. 27, 2002	Law Group LLP	1 1
(65)	Prior Publication Data	(57) ABSTRACT A system for protocol processing in a computer	addre
	Related U.S. Application Data	an intelligent network interface card (INIC) or tion processing device (CPD) associated with	C
(63)	Continuation of application No. 10/092,967, filed on Mar. 6, 2002, now Pat. No. 6,591,302, which is a	puter. The INIC provides a fast-path that avo processing for most large multi-packet messa accelerating data communication. The INIC als	Contr
(60)	(Continued) Provisional application No. 60/098,296, filed on Aug. 27, 1998, provisional application No. 60/061,809, filed on Oct. 14. 1997.	host for those message packets that are chosen ing by host software layers. A communication c for a message is defined that allows DMA cont INIC to move data, free of headers, directly t destination or source in the host. The context is	prisin
(51)	Int. Cl. <i>G06F 13/38</i> (2006.01) <i>G06F 15/17</i> (2006.01)	INIC as a communication control block (CCB passed back to the host for message processing The INIC contains specialized hardware circ much faster at their specific tasks than a gen	act
(52)	U.S. Cl	CPU. A preferred embodiment includes a trio processors with separate processors devoted	1
(58)	Field of Classification Search	receive and management processing, with full munication for four fast Ethernet nodes.	r
	See application file for complete search history.	22 Claims, 89 Drawing Sheets	t
		FAST-PATH SOURCE/DEST	i
	159	166 APPLICATION	f
	PROCESSOR S S	162 NETWORK 160 DATA LINK	t

1. A device for use with a first apparatus that is connectable to a second apparatus, the first apparatus containing a memory and a first processor operating a stack of protocol processing layers that create a context for communication, the context including a media access control (MAC) layer address, an Internet Protocol (IP) address and Transmission Control Protocol (TCP) state information, the device comprising:

communication processing mechanism connected to the first processor, said communication processing mechanism containing a second processor running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory and the TCP state information is updated by said second processor.

Ex. 1001 (036 Patent), Claim 1.

It would be obvious to use Tanenbaum96's fast-path connection records with Erickson

THIRD EDITION

TTTT

ANDREW

NFORMATIC SUPER HIGHWAY 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,

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

> Paper 2 (036 Petition) at 68-69; Ex. 1003.110 (036 Horst Decl.); Ex. 1006.584-.585 (Tanenbaum96).

Tanenbaum96: TCP state information is stored in a connection record

COMPUTER NETWOR ANDREW S. TANENBAUM

THIRD EDITION

heavy solid line) then later the path of a server (the heavy dashed line). When an application on the client machine issues a CONNECT request, the local TCP entity creates a connection record, marks it as being in the *SYN SENT* state, and sends a *SYN* segment. Note that many connections may be open (or being opened) at the same time on behalf of multiple applications, so the state is per connection and recorded in the connection record. When the *SYN+ACK* arrives, TCP sends the



Paper 2 (036 Petition) at 68-69; Ex. 1003.110-.111 (036 Horst Decl.) Ex. 1006.549 (Tanenbaum96).

67

036 Patent: Disputes

- 2. The combination of Erickson and Tanenbaum96 discloses the limitations of claims 1-7 of the 036 Patent
 - a) The prior art discloses "said communication processing mechanism containing a second processor" (all claims)
 - b) The prior art discloses "[second processor] running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory" (all claims)
 - c) The prior art discloses "the TCP state information is updated by said second processor" (all claims)
 - d) <u>The prior art discloses a "receive sequencer with directions to</u> <u>classify said packet" on "said communication processing</u> <u>mechanism" (claim 2)</u>
 - e) The prior art discloses a "receive sequencer with directions to generate a summary" on "said communication processing mechanism (claim 3)

"Receive sequencer with directions to classify said packet"



2. The device of claim 1, wherein said communication processing mechanism includes a receive sequencer with directions to classify said packet, wherein said packet contains control information corresponding to the stack of protocol layers.

Ex. 1001 (036 Patent), Claim 2.

Erickson's adapter classifies received packets by application and protocol

Ur	nited	Stat	es Patent	[19]	[11]	Patent Number:	5,768,618
Eric	ckson e	t al.			[45]	Date of Patent:	Jun. 16, 1998
[54]	METHO OF ACT COMPU VALUE SUB PO	OD FOR FIONS E UTER IN S BEINO ORTION	PERFORMING SI N DEVICE CONNE RESPONSE TO S WRITTEN INTO S OF ADDRESS SP	EQUENCE CTED TO PECIFIED SNOOPED ACE	"The D/ and Shin 1988. C Californi	ASH Local Kernal Structure Yuan Tzou, Report No. U computer Science Division ia, Berkeley 94720.	e" by David P. Anderson CB/CSD 88/463, Nov. 7 (EECS). University of
[75]	Inventor	s: Gene Hund Mulle Diego	R. Erickson; Dougl ley, both of Poway; r; Curtis H. Stehley , all of Calif.	as E. P. Keith 7. both of San	"A Usen municati National Box 200 (Aug. 15	s' Guide to PICL—A Porta on Library" By G.A. Ge Laboratory, Mathematical 99. Bldg, 9207–A. Oak F 1990).	able Instrumented Com- rist et. al., Oak Ridge Sciences Section. P.O. Ridge, TN 37831-8083
[73]	Assigne	: NCK	Corporation. Dayto	n. Onio	"Archite	cture and Implementation	of Vulcan" By Craig B.
[21]	Appl. N	o.: 577,67	1 1995		Stunkel. New Yor	et. al., IBM Research Divis k (Sep. 22, 1993).	sion, Yorktown Heights,
[51] [52] [58]	Int. CL ⁴ U.S. Cl. Field of	Search 395/	829, 832, 846, 882, 2	G06F 15/02 395/829 395/821. 823, 284, 309, 500, 473	"MPI-F: by Hub Research	An MPI Prototype Impler ertus Franke et. al., pub. Center, Yorktown Heights	nentation on IBM SP1" by IBM. T.J. Watson s, New York 10598.
[56]	,	Ref U.S. PATI	erences Cited ENT DOCUMENTS		Primary Attorney Welter 8	Examiner—Moustafa M. M. Agent, or Firm—Merchar Schmidt	deky nt, Gould. Smith. Edell.
443335555555	589,063 ,777,589 ,016,161 ,016,166 ,127,098 ,280,587 ,420,987 ,548,778 ,553,244 ,642,481 ,671,442 FOI 551148	5/1986 : 5/1991 : 5/1991 : 6/1992 : 1/1994 : 5/1995 : 8/1995 : 8/1996 : 8/1996 : 6/1997 : 9/1997 : 8/1997 : 8/1996 : 6/1997 : 7/1993 : 7/1993 : 5/1993 : 5/1995	Shab et al		[57] A metho nected te address memory prises vi UO devis the UO d the virtu and/or m detects w	ABSTRACT d of controlling an input/o a computer to facilitate fa space for the I/O device i of the computer, wherein rual registers that are used c. In essence, control regi evice are mapped into the vi al address space is back emory on the I/O device. The rites to the address space.	r utput (I/O) device con- st I/O data transfers. An is created in the virtual the address space com- d to directly control the termonory of irtual address space, and of by control registers hereafter, the I/O device is a result, a pre-defined
		OTHER	PUBLICATIONS		sequence	of actions can be triggere mine specified values into t	ed in the I/O device by the data written into the
"The Virtu David	Performa al Memor P. Ander	nce of N ry Remaj son. in So	Acssage-Passing Usi pping". by Shin-Yu ftware-Practice & E	ing Restricted an Tzou and <i>sperience</i> , vol.	mapped	virtual address space.	
21(3)	. 251-267	(Mar. 19	991).			19 Claims, 7 Drawin	ng Sheets
			THE USER PROCESSS VARIAL BLOCKS INTER- VARIAL BLOCKS INTER- PER JULIA PROCESS MOTION VARIANCE	2 N32	AND THE AND TH	NUCLEAR STATES	

Paper 2 (036 Petition) at 70; Paper 41 (036 Reply) at 18-19; Ex. 1003.112 (036 Horst Decl.); Ex. 1005 (Erickson) at 5:41-51.



Protocol scripts typically serve two functions. The first function is to describe the protocol the software application is using. This includes but is not limited to how to locate an application endpoint, and how to fill in a protocol header template from the application specific data buffer. The second function is to define a particular set of instructions to be performed based upon the protocol type. Each type of protocol will have its own script. Types of protocols include, but are not limited to, TCP/IP, UDP/IP, BYNET lightweight datagrams, deliberate shared memory, active message handler, SCSI, and File Channel

70

Tanenbaum96: Receive sequencer receives and classifies packets



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.

An optimization that often speeds up connection record lookup even more is just to maintain a pointer to the last one used and try that one first. Clark et al. (1989) tried this and observed a hit rate exceeding 90 percent. Other lookup heuristics are described in (McKenney and Dove, 1992).

The TPDU is then checked to see if it is a normal one: the state is *ESTAB*-*LISHED*, 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.

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-tomemory copy, assuming the network itself is fast enough.

Paper 2 (036 Petition) at 70-71; Paper 41 (036 Reply) at 18-19; Ex. 1003.112-.114 (036 Horst Decl.); Ex. 1006.584-.585 (Tanenbaum96).

Tanenbaum96: TCP packet contains control information

THIRD EDITION COMPUTER NET ANDREW S. TANENBAR	N WORKS					
	Source port	Destination port	VER.	IHL	TOS	Total length
	Sequenc	lde	entific	ation	Fragment offset	
	Acknowledge	ement number	TTI	L	Protocol	Header checksum
	Len Unused	Window size			Source	address
	Checksum	Urgent pointer			Destinatio	n address
BULLICOMPARTINE ALONA		a)			(b)
WELCOME TO THE INFORMATION HIGHWAY	Fig. 6-50. (a) taken from the	TCP header. (b) IP head prototype without chang	ler. In bo e.	th ca	ses, the sh	aded fields are
Emelogie	1					

Paper 2 (036 Petition) at 72; Paper 41 (036 Reply) at 18-19; Ex. 1003.114-.115 (036 Horst Decl.); Ex. 1006.584 (Tanenbaum96).
- 2. The combination of Erickson and Tanenbaum96 discloses the limitations of claims 1-7 of the 036 Patent
 - a) The prior art discloses "said communication processing mechanism containing a second processor" (all claims)
 - b) The prior art discloses "[second processor] running instructions to process a message packet such that the context is employed to transfer data contained in said packet to the first apparatus memory" (all claims)
 - c) The prior art discloses "the TCP state information is updated by said second processor" (all claims)
 - d) The prior art discloses a "receive sequencer with directions to classify said packet" on "said communication processing mechanism" (claim 2)
 - e) <u>The prior art discloses a "receive sequencer with directions to</u> <u>generate a summary" on "said communication processing</u> <u>mechanism (claim 3)</u>

) 7

"Receive sequencer with directions to generate a summary"

(12) United States Patent US 7,237,036 B2 (10) Patent No · (45) Date of Patent: Boucher et al. Jun. 26, 2007 (54) FAST-PATH APPARATUS FOR RECEIVING (56)References Cited DATA CORRESPONDING A TCP CONNECTION (75) Inventors: Laurence B. Boucher, Saratoga, CA (US); Stephen E. J. Blightman, San Jose, CA (US); Peter K. Craft, San Francisco, CA (US): David A, Higger Saratoga, CA (US); Clive M. Philbrick San Jose, CA (US); Daryl D. Starr, Milpitas, CA (US) (73) Assignee: Alacritech, Inc., San Jose, CA (US) (*) Notice: Subject to any disclaimer, the term of thi patent is extended or adjusted under 2 U.S.C. 154(b) by 672 days. (21) Appl. No.: 10/260,112 (22) Filed: Sep. 27, 2002 (65) Prior Publication Data US 2004/0073703 A1 Apr. 15, 2004 Related U.S. Application Data (63) Continuation of application No. 10/092,967, filed Mar. 6, 2002, now Pat. No. 6,591,302, which it (Continued) (60) Provisional application No. 60/098,296, filed on Aug , 1998, provisional application No. 60/061,80 filed on Oct. 14, 1997. (51) Int. Cl. passed back to the host for message processing by the host G06F 13/38 The INIC contains specialized hardware circuits that are much faster at their specific tasks than a general purpose (2006.01)G06F 15/17 (2006.01) (52) U.S. Cl. CPU. A preferred embodiment includes a trio of pipelined processors with separate processors devoted to transmit, 370/474; 370/396; 370/469 receive and management processing, with full duplex communication for four fast Ethernet nodes 707/2-4, 10; 712/19, 52 See application file for complete search history. 22 Claims, 89 Drawing Sheets FAST-PATH SOURCE/DEST 168 -APPLICATIO 166 TRANSPORT 170 162-4 NETWORK PROCESSOR HARDWARE LOGI DATA LINK

3. The device of claim **1**, wherein said communication processing mechanism includes a receive sequencer with directions to generate a summary of a second message packet received from the network, said second packet containing control information corresponding to the stack of protocol layers, and said instructions including an instruction to compare said summary with said context.

Ex. 1001 (036 Patent), Claim 3.

Tanenbaum96: Summary (IP addresses and ports) compared against context



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.

An optimization that often speeds up connection record lookup even more is just to maintain a pointer to the last one used and try that one first. Clark et al. (1989) tried this and observed a hit rate exceeding 90 percent. Other lookup heuristics are described in (McKenney and Dove, 1992).

The TPDU is then checked to see if it is a normal one: the state is *ESTAB-LISHED*, 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.

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-tomemory copy, assuming the network itself is fast enough.

Paper 2 (036 Petition) at 73-74; Paper 41 (036 Reply) at 18-19; Ex. 1003.117 (036 Horst Decl.); Ex. 1006.584-.585 (Tanenbaum96).

Tanenbaum96: TCP packet contains control information

THIRD EDITI COMPUTER NET ANDREW S. TANENBA	ON WORKS		
Andread a sile -	Source port Destination port	VER. IHL TOS	Total length
	Sequence number	Identification	Fragment offset
	Acknowledgement number	TTL Protocol	Header checksum
Trail Council of Counc	Len Unused Window size	Source	address
	Checksum Urgent pointer	Destinati	on address
	(a) (b)		(b)
WELCOME TO THE INFORMATION SUPER HIGHWAY	Fig. 6-50. (a) TCP header. (b) IP head taken from the prototype without change	ler. In both cases, the si	haded fields are
Divertey			

Paper 2 (036 Petition) at 74; Paper 41 (036 Reply) at 18-19; Ex. 1003.118 (036 Horst Decl.); Ex. 1006.584 (Tanenbaum96).

Tanenbaum96: Comparison of summary to context verifies packet is candidate for fast-path



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.

An optimization that often speeds up connection record lookup even more is just to maintain a pointer to the last one used and try that one first. Clark et al. (1989) tried this and observed a hit rate exceeding 90 percent. Other lookup heuristics are described in (McKenney and Dove, 1992).

The TPDU is then checked to see if it is a normal one: the state is *ESTAB*-*LISHED*, 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.

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-tomemory copy, assuming the network itself is fast enough.

Paper 2 (036 Petition) at 73-75; Paper 41 (036 Reply) at 18-19; Ex. 1003.117-119 (036 Horst Decl.); Ex. 1006.584-.585 (Tanenbaum96).

- 3. Motion to Amend 036 Patent should be denied
 - a) Patent Owner has improperly expanded the scope of the claims
 - b) Patent Owner does not show adequate written description support
 - c) Substitute claims are indefinite
 - d) Substitute claims are obvious

Substitute claim 23 adds an alternative to a recited step in claim 1

Original Claim 1	Substitute Claim 23
[1.P.1]. A device for use with a first apparatus that is	[23.P.1] A device for use with a first apparatus that is
connectable to a second apparatus,	connectable to a second apparatus,
[1.P.2]. the first apparatus containing a memory and a	[23.P.2] the first apparatus containing a memory and a
first processor	first processor
[1.P.3] operating a stack of protocol processing layers	[23.P.3] operating a stack of protocol processing layers
that create a context for communication, the context	that create a context for communication, the context
including a media access control (MAC) layer address,	including a media access control (MAC) layer address, an
an Internet Protocol (IP) address and Transmission	Internet Protocol (IP) address and Transmission Control
Control Protocol (TCP) state information, the device	Protocol (TCP) state information, the device comprising:
comprising:	
[1.1] a communication processing mechanism connected	[23.1] a communication processing mechanism connected
to the first processor,	to the first processor,
[1.2] said communication processing mechanism	[23.2] said communication processing mechanism
containing a second processor	containing a second processor
[1.3] running instructions	[23.3] running instructions
to process a message packet	
	on the second processor, wherein the second processor
	determining whether an incoming message packet should
	be processed by the second processor,
	[23.4] if the incoming message packet should be
	processed by the second processor, processing the
	incoming message packet, without involving the stack of
	processing protocol processing layers,
such that the context is employed to transfer data	such that the context is employed to transfer data
contained in said packet to the first apparatus memory	contained in said packet to the first apparatus memory
and	and
[1.4] the TCP state information is updated by said	[23.5] the TCP state information is updated by said
second processor.	second processor,
	[23.6] if the incoming massage nacket should not be
	processed by the second processor passing the incoming
	message packet to the first processor for further
	processing.

Requires "fast-path"

Requires either "fast path" <u>or</u> "slow path"

Paper 36 (036 Opp. to Motion to Amend) at 6.

- 3. Motion to Amend 036 Patent should be denied
 - a) Patent Owner has improperly expanded the scope of the claims
 - b) <u>Patent Owner does not show adequate written</u> <u>description support</u>
 - c) Substitute claims are indefinite
 - d) Substitute claims are obvious

PO must supply written description support after Aqua Products



Beyond that change, generally speaking, practice and procedure before the Board will not change. For example, a patent owner still must meet the requirements for a motion to amend under 37 C.F.R. § 42.121 or § 42.221, as applicable. That is, a motion to amend must set forth written description support and support for the benefit of a filing date in relation to each substitute claim, and respond to grounds of unpatentability involved in the trial. Likewise, under 37 C.F.R. § 42.11, all parties have a duty of

intel

PO identifies same 10 pages and 12 figures from original disclosure

Claims	Exemplary Support in the '112 Application	
Proposed Claim 23		
+23. A device for use with a first apparatus that is connectable to a second apparatus, the first apparatus containing a memory and a first processor operating a stack of protocol processing layers that create a context for communication, the context including a media access control (MAC) layer address, an Internet Protocol (IP) address and Transmission Control Protocol (TCP) state information, the device comprising:	See, e.g., Ex. 2020 at Abstract, Figs. 1-3, 4A-4D, and 5-12, Pages 7-8, 10-17, Cl. 1.	
a communication processing mechanism connected to the first processor,	See, e.g., Ex. 2020 at Abstract, Figs. 1-3, 4A-4D, and 5-12, Pages 7-8, 10-17, Cl. 1.	
said communication processing mechanism containing a second processor	See, e.g., Ex. 2020 at Abstract, Figs. 1-3, 4A-4D, and 5-12, Pages 7-8, 10-17, Cl. 1.	
running instructions <u>on the second processor</u> , wherein the second processor determining whether an incoming message packet should be processed by the second processor,	See, e.g., Ex. 2020 at Abstract, Figs. 1-3, 4A-4D, and 5-12, Pages 7-8, 10-17, Cl. 1.	
if the incoming message packet should be processed by the second processor, to processing the a incoming message packet, without involving the stack of processing protocol processing layers, such that the context is employed to transfer data contained in said packet to the first apparatus memory and the TCP state information is updated by said second processor,	<i>See</i> , <i>e.g.</i> , Ex. 2020 at Abstract, Figs. 1-3, 4A-4D, and 5-12, Pages 7-8, 10-17, Cl. 1	
if the incoming message packet should not be processed by the second processor, passing the incoming message packet to the first processor for further processing.	<i>See</i> , <i>e.g.</i> , Ex. 2020 at Abstract, Figs. 1-3, 4A-4D, and 5-12, Pages 7-8, 10-17, Cl. 1.	

Same written description support as 072 Patent

Paper 21 (036 Motion to Amend) Appendix A at i-ii.

Too late to provide written description support in Reply

 PO provides alleged "exemplary" written description support <u>for</u> <u>the first time</u> in its Reply

VII. <u>THE PROPOSED AMENDMENTS ARE SUPPORTED BY THE</u> <u>WRITTEN DESCRIPTION</u>

65. It is my opinion that the proposed amendments are supported by the

written description, along with the Application (No. 10/260,112) and Provisional

Application (No. 60/061,809).

Paper 42 (036 Reply ISO Motion to Amend) at 6; Ex. 2305 (Almeroth Decl. ISO Reply) at 22.

Written description support provided by PO is insufficient

- Patent Owner cites to written description support not included in its original motion (e.g., pages 18-21 of Ex. 2020)
- Patent Owner has not identified any written description support for:
 - "running instructions on the second processor, wherein the second processor determining..."
 - "such that the context [including a MAC layer address] is employed to transfer data"

Paper 50 (036 Sur-Reply for Motion to Amend) at 5-6.

- 3. Motion to Amend 036 Patent should be denied
 - a) Patent Owner has improperly expanded the scope of the claims
 - b) Patent Owner does not show adequate written description support
 - c) Substitute claims are indefinite
 - d) Substitute claims are obvious

Substitute claim 23 requires processing to determine whether to continue processing

23.3) <u>running instructions on the second processor</u>, wherein the second processor determining whether an incoming message packet **should be processed by the second processor**,

Paper 36 (036 Opp. to Motion to Amend) at 10-11.

intel

- 3. Motion to Amend 036 Patent should be denied
 - c) Substitute claims are obvious
 - i. <u>Prior art discloses "the second processor determining</u> whether an incoming message packet should be processed by second processor" (limitation 23.3)
 - ii. Prior art discloses "processing the incoming message packet [by the second processor]" (limitation 23.4)
 - iii. Prior art discloses "passing the incoming message packet to the first processor for further processing" (limitation 23.5)

Erickson discloses a second processor



Paper 36 (036 Opp. to Motion to Amend) at 13; Ex. 1005 (Erickson) at 4:18-23.

intel

Slow and fast applications can be used simultaneously

United States Patent [19]	[11] Pa
Erickson et al.	[45] D a
[54] METHOD FOR PERFORMING SEQUEN OF ACTIONS IN DEVICE CONNECTED COMPUTER IN RESPONSE TO SPECIFI VALUES BEING WRITTEN INTO SNOO SUB FORTIONS OF ADDRESS SPACE	CE "The DASH I TO and Shin-Yua IED 1988, Compu California, Be
[75] Inventors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of Diego, all of Calif.	"A Users' Gu munication L of San National Lab
[73] Assignce: NCR Corporation, Dayton, Ohi	
[21] Appl. No.: 577,678	It
[22] Fried: Dec. 21, 1995 [51] Int. CL ⁶	-
[52] U.S. Cl	the v
[56] References Cited U.S. PATENT DOCUMENTS	does
4,589,063 5/1946 Shah et al	a sta
5.420,987 5/1995 Reid et al	is. a
FOREIGN PATENT DOCUMENTS	inste
351148 7/1993 European Pat. Off OTHER PUBLICATIONS	HI YC
"The Performance of Message-Passing Using R Virtual Memory Remapping", by Shin-Yuan Ta David P. Anderson, in Software-Practice & Experies 21(2), 251-262 (Mar 1991).	dev
TWO USER RECOONSES VIETUL ASSOCIATION VIETUL ASSOCIATION	dev:
702	cess
PERJORA PROCESS SECURI OTALIA VATURA, NATIONAL	dev

US005768618A
[11] Patent Number: 5,768,618
[45] Date of Patent: Jun. 16, 1998
"The DASH Local Kernal Structure" by David P. Anderson
and Shin-Yuan Tzoz. Report No. UCB/CCB 88/463, Nov. 7,
1988. Computer Science Division (EECS). University of
California. Berkeley 94720.
"A Users' Guide to FIGL—A Portable Instrumented Communication Library" By GA. Goist et. al., Oak Ridge

ory. Mathematical Sciences Section. P.O.

"Determination" must be made between two applications

It will be recognized by those skilled in the art that using he virtual hardware implementation of the present invention loes not preclude a user process from simultaneously using a standard device driver on the same I/O device adapter. That is, a user process can use the virtual hardware of the present invention to provide a direct streamlined path to a given I/O device adapter, while standard slower access to the I/O device adapter is concurrently provided to other user processes within the same system by normal streams or other device drivers.

> Paper 36 (036 Opp. to Motion to Amend) at 14; Ex. 1005 (Erickson) at 8:65-9:7.

POSA would place header prediction on Erickson's adapter

COMPUTER NETWORKS

ANDREW S. TANENBAUM



The TPDU is then checked to see if it is a normal one: the state is *ESTAB*-*LISHED*, 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.

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

> Paper 36 (036 Opp. to Motion to Amend) at 14-15; Ex. 1006.585 (Tanenbaum96).

- 3. Motion to Amend 036 Patent should be denied
 - c) Substitute claims are obvious
 - i. Prior art discloses "the second processor determining whether an incoming message packet should be processed by second processor" (limitation 23.3)
 - ii. <u>Prior art discloses "processing the incoming message</u> packet [by the second processor]" (limitation 23.4)
 - iii. Prior art discloses "passing the incoming message packet to the first processor for further processing" (limitation 23.5)

Second processor in Erickson copies data from I/O adapter to host

United States Patent [19]	[11] Patent
Erickson et al.	[45] Date o
[54] METHOD FOR PERFORMING SEQUENCE OF ACTIONS IN DEVICE CONNECTED TO COMPUTER IN RESPONSE TO SPECIFIED VALUES BEING WRITTEN INTO SNOOPED SUB PORTIONS OF ADDRESS SPACE	"The DASH Local H and Shin-Yuan Tzou 1988, Computer Sc California, Berkeley
 [75] Inventors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of San Diego, all of Calif. [73] Assimum NCB Communic Datas Ohio. 	"A Users' Guide to munication Library National Laboratory Box 2009, Bldg, 9 (Aug, 1990).
(75) Anagaet. Her corporation, Dayton, Onto	"Architecture and Is
[21] Appl. No.: 577,678	Stunkel, et. al., IBM
[22] Filed: Dec. 21, 1995	New York (Sep. 22
[51] Int. CL ⁶	"MPI-F: An MPI F by Hubertus Frank Research Center, Y
395/829, 832, 840, 882, 284, 309, 500, 473	
[56] Beferences Cited	Primary Examiner-
U.S. DATENTE DOCUMENTS	Welter & Schmidt
U.S. PALENT DOCUMENTS	water of ovintable
4.277060 37988 3846 48. 3962 3076151 39791 Value et al. 39627 506166 57991 Value et al. 39677 506166 57991 Value et al. 39677 506166 57991 Value et al. 39677 522086 5992 Rosenhait et al. 39568 542087 5795 Rold et al. 395688 543478 8798 Rores et al. 395688 5542478 87996 Nercoss et al. 395628 5542441 67977 Poducent et al. 395628 5542441 67997 Poducent et al. 395688	[57] A method of control nected to a compute address space for t memory of the com prises virtual regists I/O device. In esset the I/O device are m
FOREIGN PATENT DOCUMENTS	the virtual address
551148 7/1993 European Pat. Off.	and/or memory on t detects writes to the
OTHER PUBLICATIONS	sequence of actions
The Performance of Message-Passing Using Restricted Virtual Memory Remapping", by Shim-Yuan Tzou and David P. Anderson. in Software-Practice d Experience, vol. 21(3), 251–257 (Mar. 1991).	programming specif mapped virtual add
TWO USER PROCESSES VIPTURE MEMORY ANTH	



FIG. 5 is a block diagram illustrating the system organization between a main memory and an I/O device adapter memory which is compatible with the present invention. The main memory 502 implementation includes a hardware register 504 and a buffer pool 506. The I/O device adapter implementation includes a software register 508 and a physical address buffer map 510 in the adapter's memory 512. An endpoint table 514 in the memory 512 is used to organize multiple memory pages for individual user processes. Each entry within the endpoint table 514 points to various protocol data 518 in the memory 512 in order to accommodate multiple communication protocols, as well as previously defined protocol scripts 516 in the memory 512, which indicate how data or information is to be transferred. from the memory 512 of the I/O device adapter to the portions of main memory 502 associated with a user process.

> Paper 36 (036 Opp. to Motion to Amend) at 17; Ex. 1005 (Erickson) at 5:52-67.



Erickson discloses fast receive and transmit

United States Patent [19] Erickson et al.

(54)	METHOD FOR PERFORMING SEQUENCE OF ACTIONS IN DEVICE CONNECTED TO COMPUTER IN RESPONSE TO SPECIFIED VALUES BEING WRITTEN INTO SNOOPED SUB PORTIONS OF ADDRESS SPACE	
[75]	Inventors: Gene R. Erickson; Douglas E. Hundley, both of Poway; P. Keith Muller; Curtis H. Stehley, both of Sa Diego, all of Calif.	
[73]	Assignce: NCR Corporation. Dayton, Ohio	
[21]	Appl. No.: 577,678	
[22]	Filed: Dec. 21, 1995	
[51] [52] [58]	Int. CL ⁶	
[56]	References Cited	
	U.S. PATENT DOCUMENTS	

4.589,063 5/1986 Shah et al 4.777,589 10/1988 Boetsner et al 5.016,161 5/1991 Van Loo et al 5.016,163 5/1991 Van Loo et al 5.127,098 6/1992 Rosensthal et al ... 5.420,087 5/1994 Shimodhine et al ... 5.420,087 5/1994 Shimodhine et al ... 395/828 395/828 395/823 395/678 395/674 711/202 5,280,587 5,420,987 5,548,778 395/880 395/830 5,42,987 5/1995 Rend et al. 5,548,778 8/1996 Hirayama 5,553,244 9/1996 Norcross et al. . 5,642,481 6/1997 Pedrizetti 5,671,442 9/1997 Feeney et al. ... 395/823 395/280 395/185.01 395/834

FOREIGN PATENT DOCUMENTS 551148 7/1993 European Pat. Off. OTHER PUBLICATIONS

"The Performance of Message-Passing Using Restricted Virtual Memory Remapping", by Shim-Yuan Tzou and David P. Anderson, in Software-Practice & Experience, vol. 21(3), 251-267 (Mar. 1991).



mapped virtual address space.

[57]

ABSTRACT

A method of controlling an input/output (I/O) device con-

nected to a computer to facilitate fast VO data transfers. An address space for the I/O device is created in the virtual

memory of the computer, wherein the address space com-

prises virtual registers that are used to directly control the

I/O device. In essence, control registers and/or memory of

the I/O device are mapped into the virtual address space, and

the virtual address space is backed by control registers

and/or memory on the I/O device. Thereafter, the I/O device

detects writes to the address space. As a result, a pre-defined

sequence of actions can be triggered in the I/O device by programming specified values into the data written into the

19 Claims, 7 Drawing Sheets

Paper 36 (036 Opp. to Motion to Amend) at 17; Ex. 1005 (Erickson) at 5:6-14, Fig. 4.

FIG. 4 is a block diagram describing a direct application interface (DAI) and routing of data between processes and an external data connection which is compatible with the present invention. Processes 402 and 404 transmit and receive information directly to and from an interconnect 410 (e.g., I/O device adapter) through the DAI interface 408. The information coming from the interconnect 410 is routed directly to a process 402 or 404 by use of virtual hardware and registers, rather than using a traditional operating system interface 406.

FIG. 4



93

intelン

- 3. Motion to Amend 036 Patent should be denied
 - c) Substitute claims are obvious
 - i. Prior art discloses "the second processor determining whether an incoming message packet should be processed by second processor" (limitation 23.3)
 - ii. Prior art discloses "processing the incoming message packet [by the second processor]" (limitation 23.4)
 - iii. <u>Prior art discloses "passing the incoming message</u> <u>packet to the first processor for further processing"</u> (limitation 23.5)

Erickson discloses use of fast and slow applications

He	itad	States Patent	t con		Botent Number	5 768 619
Erio	heen a	States Faten	L [19]	[11]	Patent Number:	5,700,010
ЕГК	KSOII e	. al.		[45]	Date of Fatent:	Jun. 10, 1998
[54]	METHO OF ACT COMPU VALUES SUB PO	D FOR PERFORMING TONS IN DEVICE CON TER IN RESPONSE TO S BEING WRITTEN INT RTIONS OF ADDRESS S	SEQUENCE NECTED TO SPECIFIED O SNOOPED SPACE	"The DA and Shin 1988, Co Californi	SH Local Kernal Structure -Yuan Tzou, Report No. U6 omputer Science Division ia. Berkeley 94720.	" by David P. Anderson CB/CSD 88/463, Nov. 7, (EECS), University of
[75]	Inventors	 Gene R. Erickson; Dou Hundley, both of Poway Muller; Curtis H. Stehl Diego, all of Calif. NCR Corporation, Day 	rglas E. 7: P. Keith ley. both of San ton. Ohio	"A Users municational Box 200 (Aug. 19	¹ Guide to PICL—A Porta on Library [®] By G.A. Ge Laboratory, Mathematical 99, Bldg, 9207–A. Oak F 90).	able Instrumented Com- tist et. al., Oak Ridge Sciences Section. P.O. Lidge, TN 37831-8083
			toni one	"Archited	cture and Implementation	of Vulcan" By Craig B.
(21)	Appl. No	Dec 21 1995		Stunkel. New Yor	et. al., IBM Research Divis k (Sep. 22, 1993).	sion, Yorktown Heights,
[51] [52] [58]	Int. CL ⁶ U.S. Cl. Field of	Search	G06F 15/02 	"MPI-F: by Hube Research	An MPI Prototype Impler rtus Franke et. al., pub. Center, Yorktown Heights	nentation on IBM SP1" by IBM, T.J. Watson 5, New York 10598.
1561		References Cited		Primary	Examiner-Moustafa M. N	deky
[50]	τ	I.S. PATENT DOCUMENT	s	Welter &	Schmidt	it, Gould, Smith, Edell,
4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,	549/063 777/589 1 016.161 016.166 1.127/098 280.587 280.587 280.587 253.244 642,481 671,442 FOR 551148 Performaal I Memori I P. Anders .251-267	5/986 Shah et al. 5/988 Bioemer et al. 5/991 Via Loo et al. 5/991 Via Loo et al. 5/991 Via Loo et al. 1/994 Shimodales et al. 1/994 Shimodales et al. 1/996 Norcess et al. 6/997 Postager et al. EEGN PATENT DOCUME 1/990 EEGN Postager et al. EEGN PATENT DOCUME 1/991 European Pat. off. 0/HER PUBLICATIONS ner of Message-Passing U. Shiho- on, in Software-Practice & (Mar. 1991).	959823 959823 959674 959674 959675 959802 95980 959802	[57] A method nected to address a memory prises vir I/O device the I/O device	ABSTRACT d of controlling an inputor a compart to facilitate fa pace for the L/O device i of the computer, wherein thal registers that are uses that are used on the output evice are mapped into the vi al address space. I trites to the address space. I action can be trigger ming specified values into of actions can be trigger ming specified values into virtual address space.	r atput (JO) device con- st IO data transfers. An s created in the virtual to directly control the the address space and d by control registers that address space, and d by control registers that address space, and d by control registers in the IO device by the data written into the ag Sheets
		VIETURE MERCER OFFICE	-30 99102	MEMORY MARTIES MAGE 296	2010 1928. 10.1000	



Slow path used after determination



Paper 36 (036 Opp. to Motion to Amend) at 21-22; Ex. 1006.583 -.584 (Tanenbaum96), Fig. 6-49.