IW 7413627

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

April 23, 2013

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF THE FILE WRAPPER AND CONTENTS OF:

APPLICATION NUMBER: ***08/533,115***
FILING DATE: ***September 25, 1995***
PATENT NUMBER: ***6,108,704***
ISSUE DATE: ***August 22, 2000***

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

P. SWAIN
Certifying Officer

PATENT

Atty. Docket No. 649-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner of Patents
and Trademarks
Washington, D.C. 20231

UTILITY APPLICATION FEE TRANSMITTAL

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): Glenn W. Hutton

For: POINT-TO-POINT INTERNET PROTOCOL

Enclosed are:

[X] 28 page(s) of specification

[X] 1 page(s) of Abstract

[X] 9 page(s) of claims

[X] 6 sheets of drawings [ ] formal [X] informal

[X] 5 page(s) of Declaration and Power of Attorney

[ ] An Assignment of the invention to

[ ] Certified copy of applications

Country	Appln. No.	Filed

from which priority under Title 35 United States Code, § 119 is claimed

[ ] is enclosed.

[ ] will follow.

CERTIFICATION UNDER 37 C.F.R. § 1.10

I hereby certify that this New Application Transmittal and the documents referred to as enclosed therein are being deposited with the United States Postal Service on this date September 25, 1995 in an envelope as "Express Mail Post Office to Addressee" Mail Label Number EM302799414US addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Wendy Greenseich
(Type or print name of person mailing paper)

Wendy Greenseich
(Signature of person mailing paper)

CALCULATION OF UTILITY APPLICATION FEE

For	Number Filed			Number Extra	Rate		Basic Fee $730.00
Total Claims*	20	-20	=	0	x $22.00		$
Independent Claims	6	-3	=	3	x $76.00		$228.00
Multiple Dependent Claims	[ ] yes			Add'l. Fee	$240.00		$
	[X] no			Add'l. Fee	None	=	$
						TOTAL	$ 958.00

[X] Verified Statement of "Small Entity" Status Under 37 C.F.R. § 1.27. Reduced fees under 37 C.F.R. § 1.9(f) (50% of total) paid herewith $479.00.

[ ] The amount of $40.00 for recording the attached Assignment is included in the enclosed check.

[X] A check in the amount of $479.00 to cover the [ ] recording, [X] filing fee(s) is attached.

[ ] Charge fee to Deposit Account No. 04-1121. Order No. _____ TWO (2) COPIES OF THIS SHEET ARE ENCLOSED.

[X] Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and 1.17, at any time during the pendency of this application, or credit any overpayment of such fee(s) to Deposit Account No. 04-1121. Also, in the event any extensions of time for responding are required for the pending application(s), please treat this paper as a petition to extend the time as required and charge Deposit Account No. 04-1121 therefor. TWO (2) COPIES OF THIS SHEET ARE ENCLOSED.

Date: September 25, 1995

SIGNATURE OF ATTORNEY
Joseph J. Catanzaro
Reg. No. 25,837

DILWORTH & BARRESE
333 Earle Ovington Blvd.
Uniondale, NY 11553
Tel. No. (516) 228-8484
Fax. (516) 228-8516

*Includes all independent and single dependent claims and all claims referred to in multiple claims. See 37 C.F.R. § 1.75(c).

649-2
SHEET 1 OF 6

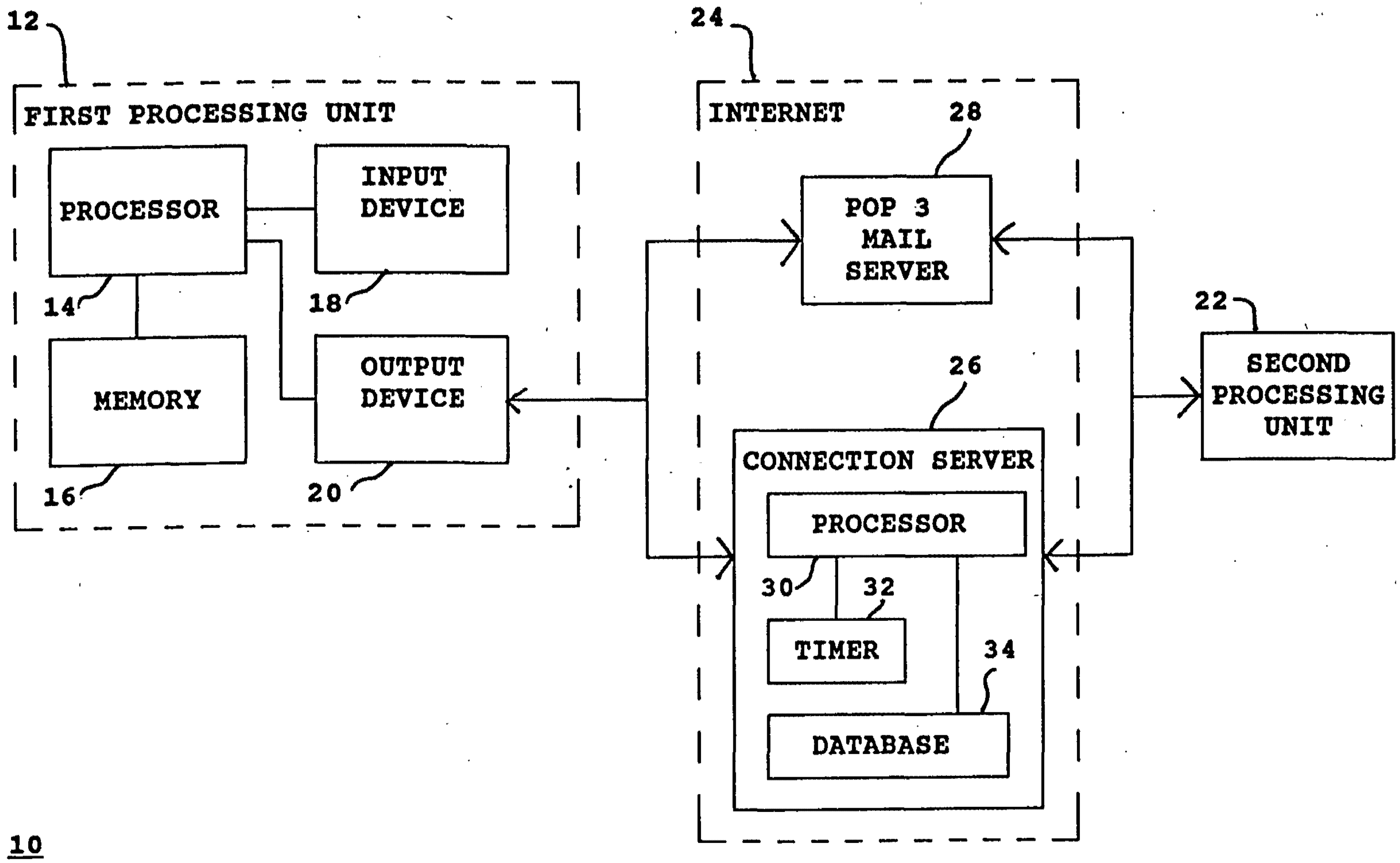


FIG. 1

649-2
SHEET 2 OF 6

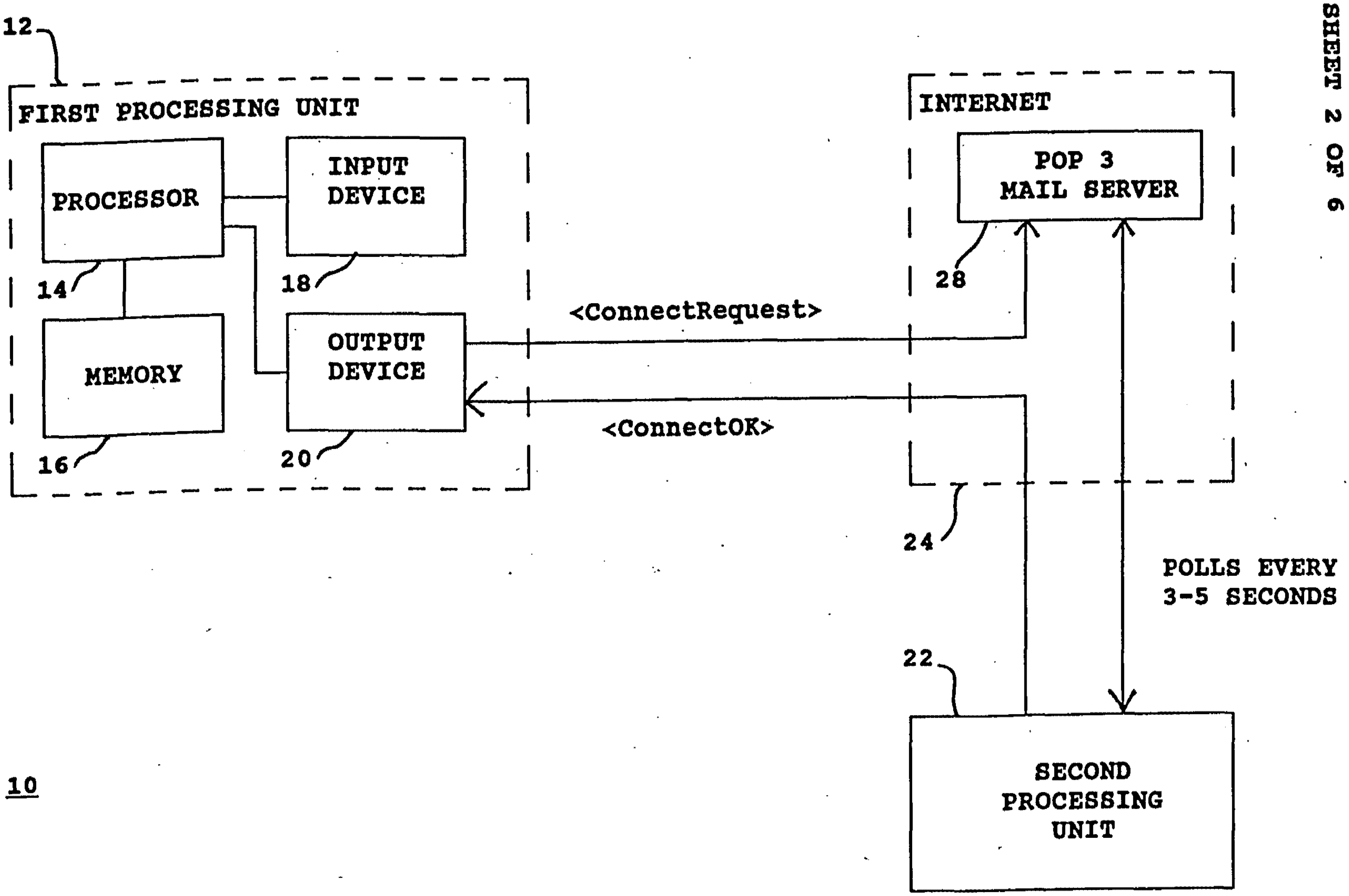


FIG. 2

08 533115

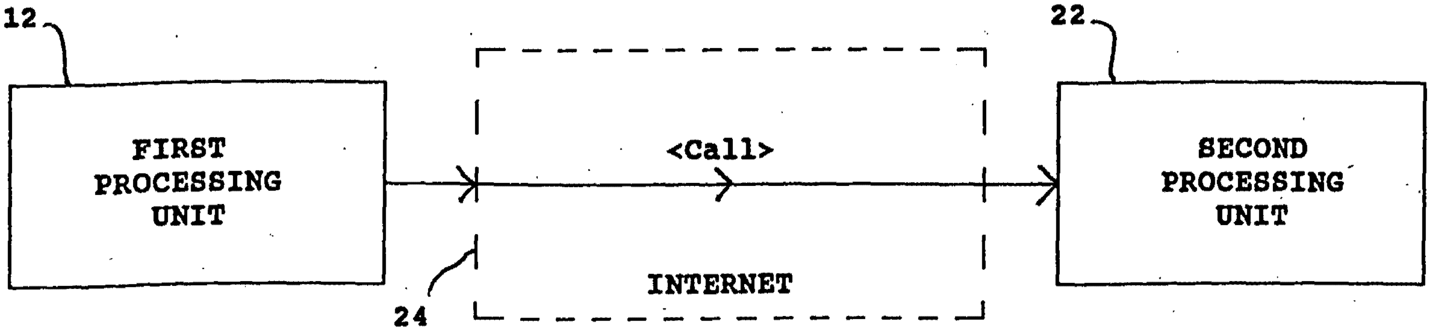

12
FIRST PROCESSING UNIT
<Call>
INTERNET
24
22
SECOND PROCESSING UNIT

FIG. 3

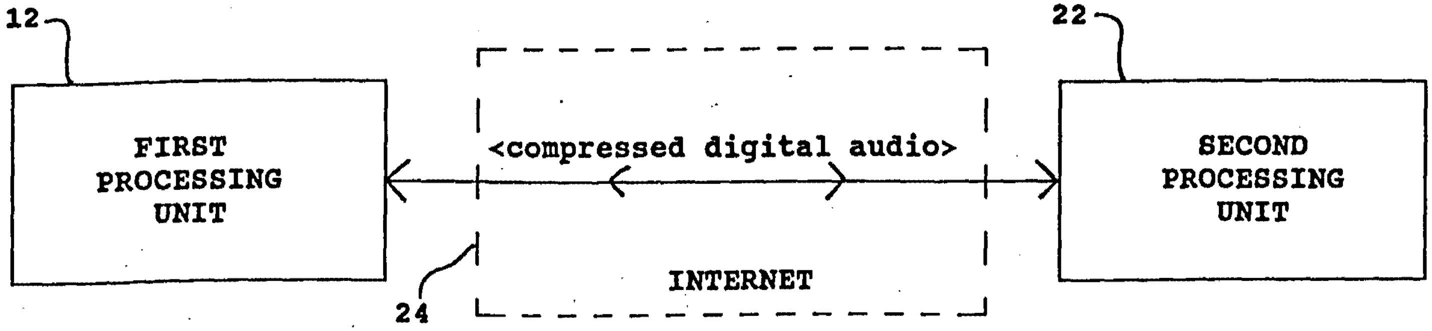

12
FIRST PROCESSING UNIT
<compressed digital audio>
INTERNET
24
22
SECOND PROCESSING UNIT

FIG. 4

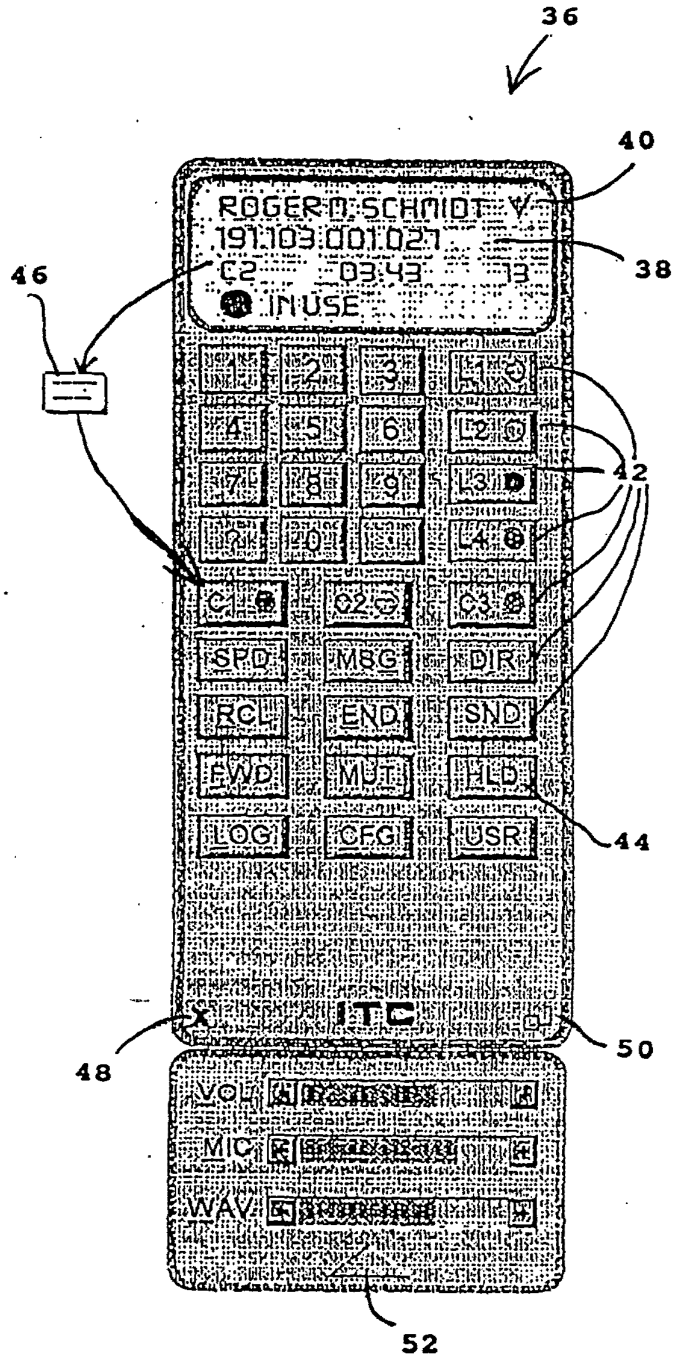

36
40
ROGER M. SCHMIDT
191.103.001.027
C2 03:43 13
IN USE
38
46
1
2
3
L1
4
5
6
L2
7
8
9
L3
?
0
L4
42
C1
C2
C3
SPD
MSG
DIR
RCL
END
SND
FWD
MUT
HLD
LOG
CFG
USR
44
X
ITC
48
50
VOL
MIC
WAV
52

FIG. 5

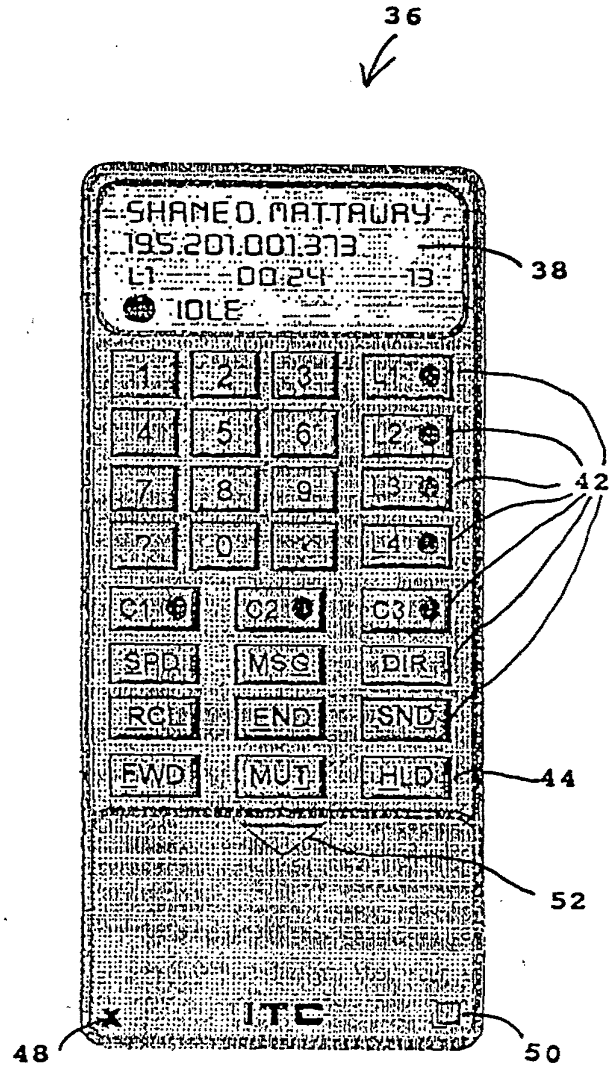

36
SHANE D. MATTAWAY
195.201.001.373
L1 00:24 13
IDLE
38
1
2
3
L1
4
5
6
L2
7
8
9
L3
?
0
L4
42
C1
C2
C3
SPD
MSG
DIR
RCL
END
SND
FWD
MUT
HLD
44
52
X
ITC
48
50

FIG. 6

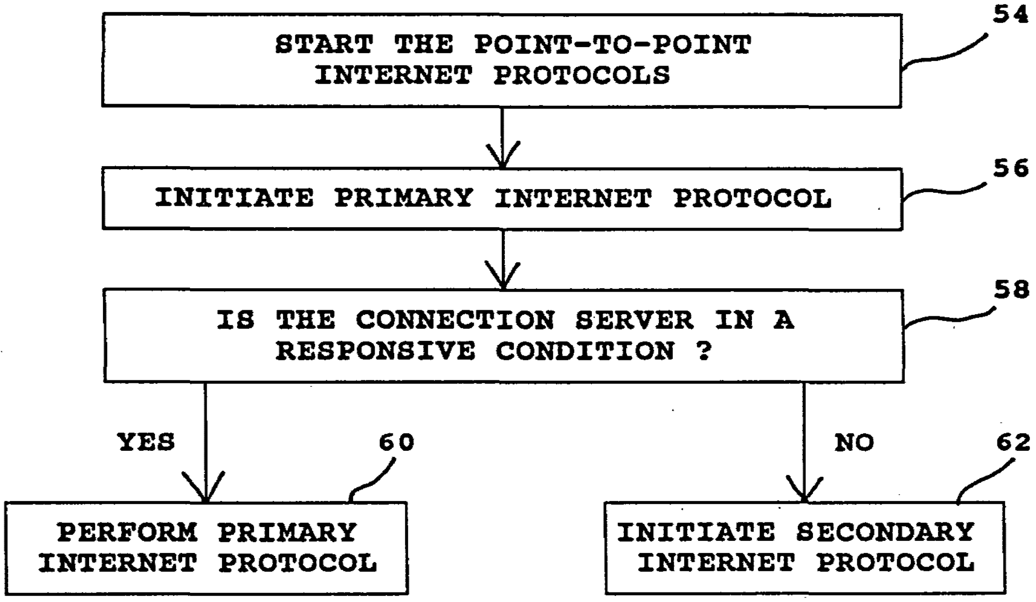

START THE POINT-TO-POINT INTERNET PROTOCOLS
54
INITIATE PRIMARY INTERNET PROTOCOL
56
IS THE CONNECTION SERVER IN A RESPONSIVE CONDITION ?
58
YES
60
NO
62
PERFORM PRIMARY INTERNET PROTOCOL
INITIATE SECONDARY INTERNET PROTOCOL

FIG. 7

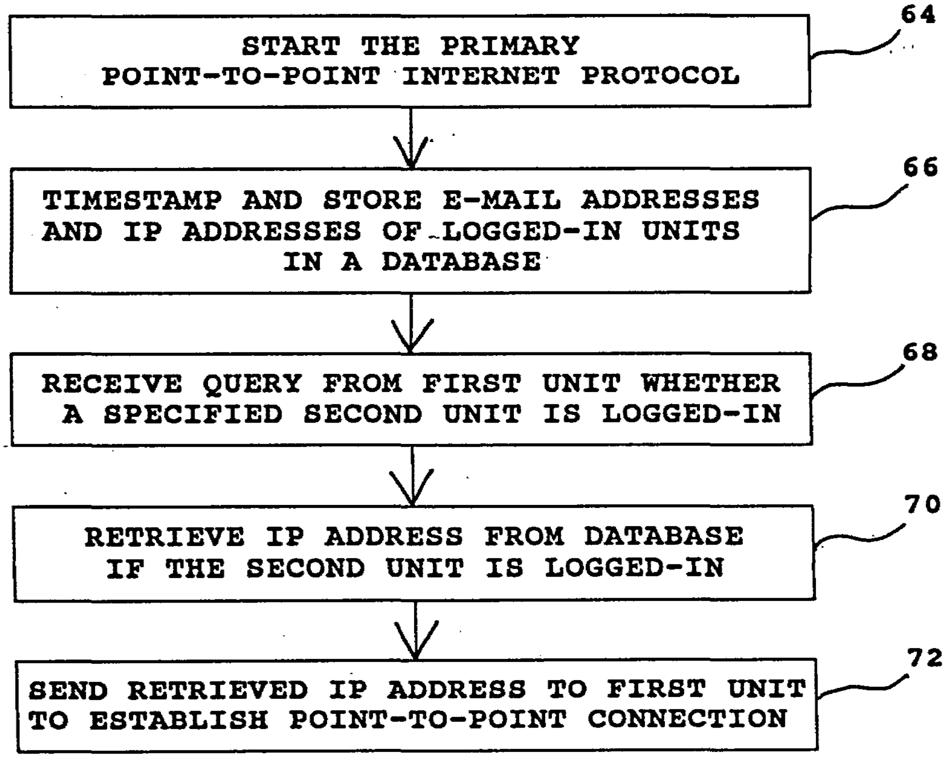

START THE PRIMARY POINT-TO-POINT INTERNET PROTOCOL
64
TIMESTAMP AND STORE E-MAIL ADDRESSES AND IP ADDRESSES OF LOGGED-IN UNITS IN A DATABASE
66
RECEIVE QUERY FROM FIRST UNIT WHETHER A SPECIFIED SECOND UNIT IS LOGGED-IN
68
RETRIEVE IP ADDRESS FROM DATABASE IF THE SECOND UNIT IS LOGGED-IN
70
SEND RETRIEVED IP ADDRESS TO FIRST UNIT TO ESTABLISH POINT-TO-POINT CONNECTION
72

FIG. 8

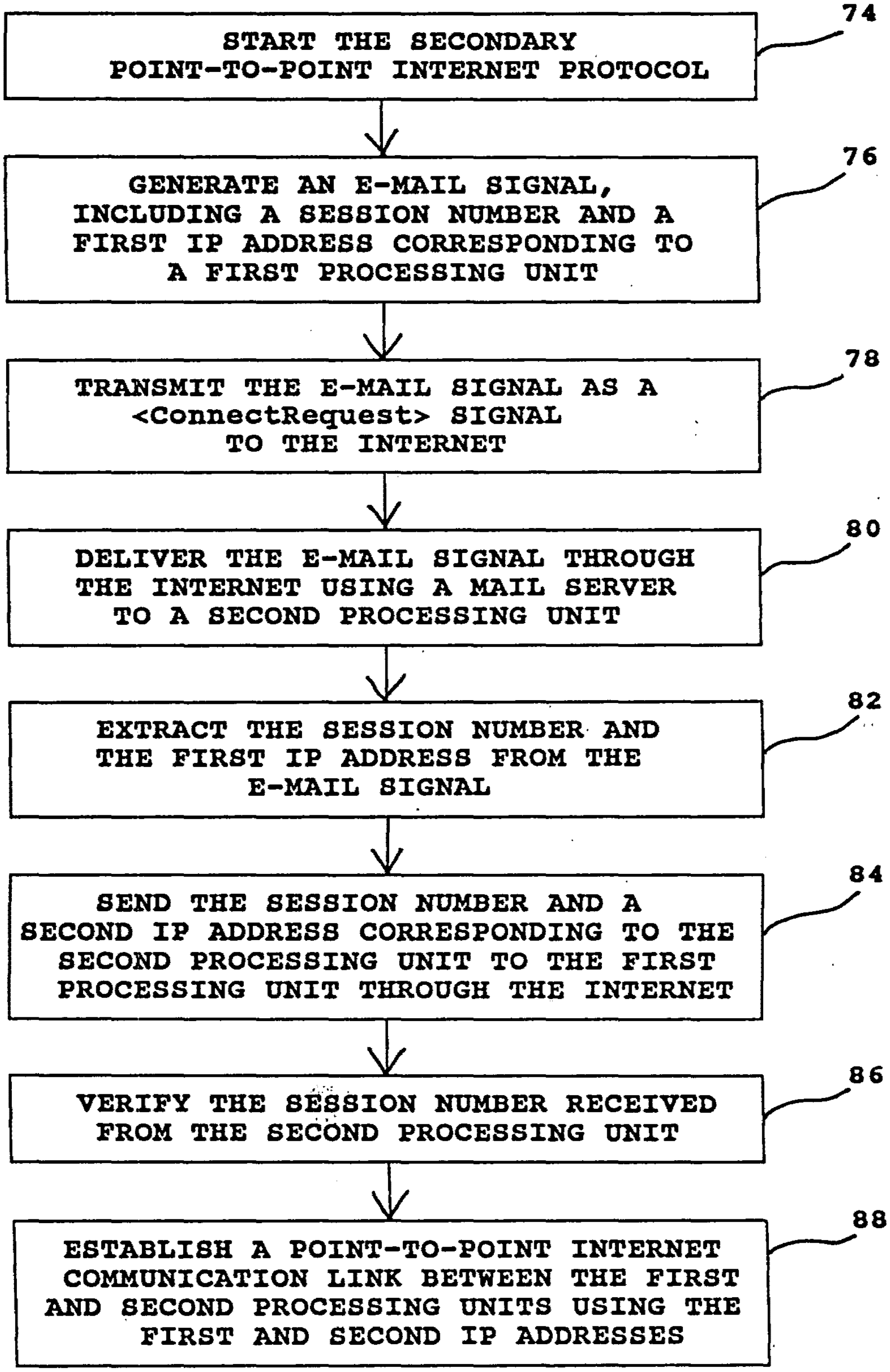

START THE SECONDARY POINT-TO-POINT INTERNET PROTOCOL
74
GENERATE AN E-MAIL SIGNAL, INCLUDING A SESSION NUMBER AND A FIRST IP ADDRESS CORRESPONDING TO A FIRST PROCESSING UNIT
76
TRANSMIT THE E-MAIL SIGNAL AS A <ConnectRequest> SIGNAL TO THE INTERNET
78
DELIVER THE E-MAIL SIGNAL THROUGH THE INTERNET USING A MAIL SERVER TO A SECOND PROCESSING UNIT
80
EXTRACT THE SESSION NUMBER AND THE FIRST IP ADDRESS FROM THE E-MAIL SIGNAL
82
SEND THE SESSION NUMBER AND A SECOND IP ADDRESS CORRESPONDING TO THE SECOND PROCESSING UNIT TO THE FIRST PROCESSING UNIT THROUGH THE INTERNET
84
VERIFY THE SESSION NUMBER RECEIVED FROM THE SECOND PROCESSING UNIT
86
ESTABLISH A POINT-TO-POINT INTERNET COMMUNICATION LINK BETWEEN THE FIRST AND SECOND PROCESSING UNITS USING THE FIRST AND SECOND IP ADDRESSES
88

FIG. 9

08 533115

729	227
Class	Subclass

UTILITY SERIAL NUMBER 08

SERIAL NUMBER

APPLICANTS

VERIFIED

FOREIGN

VERIFIED

FOREIGN

Foreign priority clair
35 USC 119 conditi
Verified and Acknow

ADDRESS

PARTS OF APPL
FILED SEPARAT

NOTICE OF ALL

5-

ISSU

Amount Due
$605.00

La
Ar

Form PTO-436A
(Rev. 8/92)

Fo

Staple Issue Slip Here

POSITION		ID NO.	DATE
CLASSIFIER		5	10-17-95
EXAMINER		300	10-17-95
TYPIST		710	10/17
VERIFIER		76	10-17
CORPS CORR.			
SPEC. HAND			
FILE MAINT.			
DRAFTING			

INDEX OF CLAIMS

Claim		Date							
Final	Original	5/24/94	4/95	7/96	10/98	5/99			
	1	✓	+	+	✓				
	2	✓	+	+	✓				
	3	✓	+	+	✓				
	4	✓			✓				
	5	✓			N	N			
	6	✓			✓				
	7	✓			✓				
	8	✓			✓				
	9	✓			✓				
	10	✓			✓				
	11	✓			✓				
	12	✓			N				
	13	✓			N				
	14	✓			N				
	15	✓			N				
	16	✓			N				
	17	✓			N				
	18	✓			N				
	19	✓			N				
	20	✓			N				
1	21	✓			✓	=			
	22	✓			N				
2	23	✓			✓	=			
3	24	✓			✓	=			
	25	✓			N				
4	26	✓			✓	=			
5	27	✓			✓	=			
6	28	✓			✓	=			
7	29	✓			✓				
8	30	✓			✓				
9	31	✓			✓				
10	32	✓			✓				
11	33	✓			✓				
12	34	✓			✓				
13	35	✓			✓				
14	36	✓			✓				
15	37	✓			✓				
16	38	✓			✓				
17	39	✓			✓				
18	40	✓			✓				
19	41	✓			✓				
20	42	✓			✓				
21	43	✓			✓				
22	44	✓			✓				
23	45	✓			✓				
24	46	✓			✓				
25	47	✓			✓				
26	48	✓	+	+	✓	=			
27	49	✓	+	+	✓	=			
28	50	✓	+	+	✓	=			

Claim		Date							
Final	Original	6/2	4/96	7/96	6/98	5/99			
29	51	✓	+	+	✓	=			
30	52	✓	+	+	✓	=			
31	53	✓	+	+	✓	=			
32	54				✓				
33	55				✓				
34	56				✓				
35	57				✓				
36	58				✓				
37	59				✓				
38	60				✓				
39	61				✓				
40	62				✓	=			
41	63				✓	=			
42	64				✓	=			
	65				N				
43	66		+	+	✓	=			
44	67		+	+	✓	=			
	68		+	+	N				
	69								
	70								
	71								
	72								
	73								
	74								
	75								
	76								
	77								
	78								
	79								
	80								
	81								
	82								
	83								
	84								
	85								
	86								
	87								
	88								
	89								
	90								
	91								
	92								
	93								
	94								
	95								
	96								
	97								
	98								
	99								
	100								

SYMBOLS

✓	Rejected
=	Allowed
. (Through numeral)	Canceled
+	Restricted
N	Non-elected
I	Interference
A	Appeal
O	Objected

(LEFT INSIDE)

35 Rule 312 Amdt 3-8-00

36. 6-28-00 7-14-99

Formal Drawings(6 shts)set

SEARCHED

Class	Sub.	Date	Exmr.
395 395	200.01 200.02 200.09 200.11 200.15	5/22/97	Pk
395 395	200.34 200.35 200.47 200.48 200.57 200.58 200.75	10/22/98 10/22/98	uR R
709 709	204 205 217 218 227 228 235	5/18/99 5/18/99	uR uR

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.
709	227	5/18/99	uR
709	204	5/18/99	R

SEARCH NOTES

	Date	Exmr.
Discussion w/ Rumbaugh re: URL & Internet references - no on APS - got W3 docs	5/22/97	FA
See APS Search on Internet	5/16/97	Pk
APS search attached	10/26/98	uR

(RIGHT OUTSIDE)

ISSUE CLASSIFICATION		
08 533115	709	227
	Class	Subclass

73665

6108704

UTILITY SERIAL NUMBER 08 533115	PATENT DATE AUG 22 2000	PATENT NUMBER PAID

SERIAL NUMBER	FILING DATE	CLASS	SUBCLASS	GROUP ART UNIT	EXAMINER
		709	227	2756	F773

APPLICANTS

Arayon

Norge Pb

Prest

Foreign priority claimed 35 USC 119 conditions met Verified and Acknowledged	□ yes □ no □ yes ☑ no Examiner's initials	AS FILED →	STATE OR COUNTRY	SHEETS DRWGS.	TOTAL CLAIMS	INDEP. CLAIMS	FILING FEE RECEIVED	ATTORNEY'S DOCKET NO.

ADDRESS

Kudirka & Jobse, LLP
Two Center Plaza
Boston, MA 02108

U.S. DEPT. OF COMM./PAT. &

PARTS OF APPLICATION FILED SEPARATELY

Applications Examiner

NOTICE OF ALLOWANCE MAILED		CLAIMS ALLOWED	
5-25-99	Assistant Examiner	Total Claims 44	Print Claim 1

ISSUE FEE			DRAWING		
Amount Due $605.00	Date Paid 8/3/99	Mark H. Rinehart Primary Examiner Primary Examiner	Sheets Drwg. 6	Figs. Drwg. 9	Print Fig. 9
Label Area		PREPARED FOR ISSUE	ISSUE BATCH NUMBER T-56		

WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 8/92)

ISSUE FEE IN FILE

PATENT NUMBER

APPLICATION SERIAL NUMBER
08/583115

APPLICANT'S NAME (PLEASE PRINT)
Hutton

REISSUE, ORIGINAL PATENT NUMBER

INTERNATIONAL CLASSIFICATION

G	0	6	F		13 / 38	
G	0	6	F		15 / 17	

ORIGINAL CLASSIFICATION

CLASS	SUBCLASS
709	227

CROSS REFERENCE(S)

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
709	204			

GROUP ART UNIT
2756

ASSISTANT EXAMINER (PLEASE STAMP OR PRINT FULL NAME)

PRIMARY EXAMINER (PLEASE STAMP OR PRINT FULL NAME)
Mark H. Rinehart
Primary Examiner

PTO 270

ISSUE CLASSIFICATION SLIP

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 1994

Application or Docket Number

533115

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE	FEE		RATE	FEE
BASIC FEE				365.00	OR		730.00
TOTAL CLAIMS	20 minus 20 =	*	x$11=		OR	x$22=	
INDEPENDENT CLAIMS	6 minus 3 =	* 3	x38=	114—	OR	x76=	
MULTIPLE DEPENDENT CLAIM PRESENT			+120=		OR	+240=	
			TOTAL	479—	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 53	Minus	** 20	= 33	x$11=	363 pd	OR	x$22=	
Independent	* 12	Minus	*** 6	= 6	x38=	234 pd	OR	x76=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+120=		OR	+240=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 68	Minus	** 53	= 15	x$11=	165.	OR	x$22=	
Independent	* 19	Minus	*** 12	= 7	x~~38~~= 41	287.	OR	x76=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+120=		OR	+240=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
C	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 58	Minus	** 68	= —	x$11=		OR	x$22=	
Independent	* 16	Minus	*** 19	= —	x38=		OR	x76=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+120=		OR	+240=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

PATENT APPLICATION SERIAL NO. 08 533115

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

280 MM 10/17/95 08533115 1 201 479.00 CK 649-2

PTO-1556
(5/87)

BAR CODE LABEL

U.S. PATENT APPLICATION

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT
08/533,115	09/25/95	395	2305

APPLICANT

GLENN W. HUTTON, MIAMI, FL.

CONTINUING DATA********************
VERIFIED

____

FOREIGN/PCT APPLICATIONS**********
VERIFIED

____

FOREIGN FILING LICENSE GRANTED 10/17/95 ***** SMALL ENTITY *****

STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS	FILING FEE RECEIVED	ATTORNEY DOCKET NO.
FL	6	20	6	$479.00	649-2

ADDRESS

JOSEPH J CATANZARO
DILWORTH & BARRESE
333 EARLE OVINGTON BLVD
UNIONDALE NY 11553

TITLE

POINT-TO-POINT INTERNET PROTOCOL

This is to certify that annexed hereto is a true copy from the records of the United States Patent and Trademark Office of the application which is identified above.

By authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

Date

Certifying Officer

Bib Data Sheet

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
08/533,115	09/25/1995 **RULE** _	709	2756	649-2

APPLICANTS

GLENN W. HUTTON, MIAMI, FL ;
SHANE D. MATTAWAY, BOCA RATON, FL ;
CRAIG B. STRICKLAND, TAMARAC, FL ;

** **CONTINUING DATA** **************************

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** SMALL ENTITY **
** 10/17/1995

		STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed	□ yes □ no	FL	6	20	6
35 USC 119 (a-d) conditions met	□ yes □ no □ Met after Allowance				
Verified and Acknowledged	Examiner's Signature Initials				

ADDRESS

21127

TITLE

POINT-TO-POINT INTERNET PROTOCOL

FILING FEE RECEIVED 2125	FEES: Authority has been given in Paper No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ______________ □ Credit

PATENT

Attorney's Docket No. 649-2

COMBINED DECLARATION AND POWER OF ATTORNEY

(ORIGINAL, DESIGN, NATIONAL STAGE OF PCT, SUPPLEMENTAL, DIVISIONAL, CONTINUATION OR CIP)

As a below named inventor, I hereby declare that:

TYPE OF DECLARATION

This declaration is of the following type: *(check one applicable item below)*

☒ original
□ design
□ supplemental

NOTE: If the declaration is for an International Application being filed as a divisional, continuation or continuation-in-part application do not check next item; check appropriate one of last three items.

□ national stage of PCT

NOTE: If one of the following 3 items apply then complete and also attach ADDED PAGES FOR DIVISIONAL, CONTINUATION OR CIP.

□ divisional
□ continuation
□ continuation-in-part (CIP)

INVENTORSHIP IDENTIFICATION

WARNING: If the inventors are each not the inventors of all the claims an explanation of the facts, including the ownership of all the claims at the time the last claimed invention was made, should be submitted.

My residence, post office address and citizenship are as stated below next to my name, I believe I am the original, first and sole inventor *(if only one name is listed below)* or an original, first and joint inventor *(if plural names are listed below)* of the subject matter which is claimed and for which a patent is sought on the invention entitled:

TITLE OF INVENTION

POINT-TO-POINT INTERNET PROTOCOL

SPECIFICATION IDENTIFICATION

the specification of which: *(complete (a), (b) or (c))*

(a) ☒ is attached hereto.

(b) □ was filed on ______________ as □ Serial No. 0 /______
or □ Express Mail No., as Serial No. not yet known ________________ and was amended on __________ ____________ *(if applicable)*.

NOTE: Amendments filed after the original papers are deposited with the PTO which contain new matter are not accorded a filing date by being referred to in the declaration. Accordingly, the amendments involved are those filed with the application papers or, in the case of a supplemental declaration, are those amendments claiming matter not encompassed in the original statement of invention or claims. See 37 C.F.R. 1.67.

(c) □ was described and claimed in PCT International Application No. ______________ filed on __________ and as amended under PCT Article 19 on ______________ *(if any)*.

ACKNOWLEDGEMENT OF REVIEW OF PAPERS AND DUTY OF CANDOR

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information

□ which is material to patentability as defined in 37, Code of Federal Regulations, §1.56

(also check the following items, if desired)

□ and which is material to the examination of this application, namely, information where there is a substantial likelihood that a reasonable examiner would consider it important in deciding whether to allow the application to issue as a patent, and

□ In compliance with this duty there is attached an information disclosure statement in accordance with 37 C.F.R. 1.98.

PRIORITY CLAIM (35 U.S.C. §119)

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed.

(complete (d) or (e))

(d) ☒ no such applications have been filed.

(e) □ such applications have been filed as follows.

NOTE: Where item (c) is entered above and the International Application which designated the U.S. itself claimed priority check item (e), enter the details below and make the priority claim.

A. PRIOR FOREIGN/PCT APPLICATION(S) FILED WITHIN 12 MONTHS (6 MONTHS FOR DESIGN) PRIOR TO THIS APPLICATION AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. §119

COUNTRY (OR INDICATE IF PCT)	APPLICATION NUMBER	DATE OF FILING *(day, month, year)*	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			□ YES □ NO
			□ YES □ NO
			□ YES □ NO
			□ YES □ NO
			□ YES □ NO

ALL FOREIGN APPLICATION(S), IF ANY FILED MORE THAN 12 MONTHS (6 MONTHS FOR DESIGN) PRIOR TO THIS U.S. APPLICATION

NOTE: If the application filed more than 12 months from the filing date of this application is a PCT filing forming the basis for this application entering the United States as (1) the national stage, or (2) a continuation, divisional, or continuation-in-part, then also complete ADDED PAGES TO COMBINED DECLARATION AND POWER OF ATTORNEY FOR DIVISIONAL, CONTINUATION OR CIP APPLICATION for benefit of the prior U.S. or PCT application(s) under 35 U.S.C. §120.

POWER OF ATTORNEY

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(List name and registration number)*

22

PETER G. DILWORTH, Reg. No. 26,450; **ROCCO S. BARRESE**, Reg. No. 25,253; **JOSEPH J. CATANZARO**, Reg. No. 25,837; **DAVID M. CARTER**, Reg. No. 30,949; **PAUL J. FARRELL**, Reg. No. 33,494; **PETER DELUCA**, Reg. No. 32,978; **ADRIAN T. CALDERONE**, Reg. No. 31,746; **GEORGE M. KAPLAN**, Reg. No. 28,375; **JEFFREY S. STEEN**, Reg. No. 32,063; **JOSEPH W. SCHMIDT**, Reg. No. 36,920; **RAYMOND E. FARRELL**, Reg. No. 34,816; **RUSSELL R. KASSNER**, Reg. No. 36,183; **FRANK CHAU**, Reg. No. 34,136; **SCOTT D. WOFSY**, Reg. No. 35,413; **ANTHONY J. NATOLI**, Reg. No. 36,223; **MICHAEL P. DILWORTH**, Reg. No. 37,311; **RICHARD F. JAWORSKI**, Reg. No. 33,515; **DANIEL E. TIERNEY**, Reg. No. 33,461, **WALTER M. EGBERT, III**, Reg. No. 37,317, **JEAN CHUNG**, Reg. No. 38,674, **CHRISTOPHER G. TRAINOR**, Reg. No. 39,517; and **GLENN F. SAVIT**, Reg. No. 37,437, each of them of **DILWORTH & BARRESE**, 333 Earle Ovington Boulevard, Uniondale, New York 11553.

SEND CORRESPONDENCE TO:	DIRECT TELEPHONE CALLS TO: *(Name and telephone number)*
JOSEPH J. CATANZARO DILWORTH & BARRESE 333 Earle Ovington Boulevard Uniondale, New York 11553	(516) 228-8484

DECLARATION

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

SIGNATURE(S)

NOTE: Carefully indicate the family (or last) name as it should appear on the filing receipt and all other documents.

1-00

Full name of **sole or first inventor** **Glenn W. Hutton**

✓Inventor's signature

✓Date 09-23-95 Country of Citizenship **Canada**

Residence **Miami, Florida** FL

Post Office Address **9725 Hammocks Boulevard, #206, Miami, Florida 33196**

Full name of **second joint inventor**, if any

Inventor's signature
Date Country of Citizenship
Residence
Post Office Address

Full name of **third joint inventor**, if any

Inventor's signature
Date Country of Citizenship
Residence
Post Office Address

Full name of **fourth joint inventor**, if any

Inventor's signature
Date Country of Citizenship
Residence
Post Office Address

Full name of **fifth joint inventor**, if any

Inventor's signature
Date Country of Citizenship
Residence
Post Office Address

CHECK PROPER BOX(ES) FOR ANY OF THE FOLLOWING ADDED PAGE(S) WHICH FORM A PART OF THIS DECLARATION

□ Signature for subsequent joint inventors.
Number of pages added _________.

□ Signature by administrator(trix), executor(trix) or legal representative for deceased or incapacitated inventor.
Number of pages added _________.

□ Signature for inventor who refuses to sign or cannot be reached by person authorized under 37 C.F.R. §1.47.
Number of pages added _________.

***

□ Added pages to combined declaration and power of attorney for divisional, continuation, or continuation-in-part (CIP) application.
Number of pages added _________.

***

□ Authorization of attorney(s) to accept and follow instructions from representative.

***

If no further pages form a part of this Declaration then end this Declaration with this page and check the following item.

☒ This declaration ends with this page.

Applicant or Patentee: Glenn Hutton

Serial or Patent No.: Not Yet Assigned

Filed or Issued: Concurrently Herewith

For: POINT-TO-POINT INTERNET PROTOCOL

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) and 1.27(b)) - INDEPENDENT INVENTOR

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office with regard to the invention entitled

POINT-TO-POINT INTERNET PROTOCOL

described in

☒ the specification filed herewith.

□ application serial no. ________, filed ________________.

□ patent no. ___________, issued ____________________.

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, convey, or license any rights in the invention is listed below:

□ no such person, concern, or organization

□ persons, concerns or organizations listed below*

NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27).

FULL NAME ________________________________

ADDRESS ________________________________
□ Individual □ Small Business Concern □ Nonprofit Organization

FULL NAME ________________________________

ADDRESS ________________________________
□ Individual □ Small Business Concern □ Nonprofit Organization

FULL NAME ________________________________

ADDRESS ________________________________
□ Individual □ Small Business Concern □ Nonprofit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Glenn Hutton
Name of inventor

✓
Signature of inventor

✓ Date 09-23-95

Name of inventor

Signature of inventor

Date

Name of inventor

Signature of inventor

Date

ABSTRACT

A point-to-point Internet protocol exchanges Internet Protocol (IP) addresses between processing units to establish a point-to-point communication link between the processing units through the Internet. A first point-to-point Internet protocol includes the steps of (a) storing in a database a respective IP address of a set of processing units that have an on-line status with respect to the Internet; (b) transmitting a query from a first processing unit to a connection server to determine the on-line status of a second processing unit; and (c) retrieving the IP address of the second unit from the database using the connection server, in response to the determination of a positive on-line status of the second processing unit, for establishing a point-to-point communication link between the first and second processing units through the Internet. A second point-to-point Internet protocol includes the steps of (a) transmitting an E-mail signal, including a first IP address, from a first processing unit; (b) processing the E-mail signal through the Internet to deliver the E-mail signal to a second processing unit; and (c) transmitting a second IP address to the first processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet.

08/533118

649-2

WHAT IS CLAIMED IS:

1. A method for establishing point-to-point Internet communication comprising the steps of:

(a) storing in a database a respective Internet Protocol (IP) address of a set of processing units that have an on-line status with respect to the Internet;

(b) transmitting a query from a first processing unit to a connection server to determine the on-line status of a second processing unit; and

(c) retrieving the IP address of the second unit from the database using the connection server, in response to the determination of a positive on-line status of the second processing unit, for establishing a point-to-point communication link between the first and second processing units through the Internet.

2. The method of claim 1 wherein the step (b) of transmitting the query includes the step of:

(b1) transmitting the query to the connection server operatively connected to the database and the Internet; and

wherein the step (c) of retrieving the IP address includes the steps of:

(c1) searching the database using the connection server;

(c2) determining the on-line status of the second processing unit;

(c3) retrieving the IP address of the second processing unit in response to the positive on-line status of the second processing unit; and

(c4) transmitting the IP address of the second processing unit to the first processing unit for establishing the point-to-point communication link between the first and second processing units through the Internet.

3. The method of claim 2 further comprising, after step (c2), the steps of:

(c5) generating an off-line message in response to a negative on-line status of the second processing unit; and

(c4) transmitting the off-line message to the first processing unit.

4. The method of claim 1 further comprising the step of:

(d) performing a secondary communication protocol in response to a non-responsive condition of the connection server.

~~5. The method of claim 4 wherein the step (d) of performing the secondary communication protocol includes the steps of:~~

~~(d1) transmitting an E-mail signal, including a first IP address, from the first processing unit;~~

~~(d2) processing the E-mail signal through the Internet to deliver the E-mail signal to the second processing unit; and~~

~~(d3) transmitting a second IP address to the first processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet.~~

~~6. An apparatus comprising:~~

~~a first processing unit including:~~

~~a program stored in a memory for performing a point-to-point Internet protocol; and~~

~~a processor for executing the point-to-point Internet protocol program to generate a query to receive an Internet Protocol (IP) address of a second processing unit, for transmitting the query through the Internet to a connection server for determining an on-line status of a second processing unit to the connection server, and for establishing a point-to-point communication link to the second processing unit using the IP address.~~

7. A system for point-to-point communications over the Internet comprising:

a database for storing a set of Internet Protocol (IP) addresses of at least one processing unit that has on-line status with respect to the Internet;

a first processing unit including:

a first program for performing a first point-to-point Internet protocol; and

a first processor for executing the first program and for transmitting a query;

a connection server, responsive to the query, for determining the on-line status of a second processing unit by searching the database, and for transmitting an on-line message to the first processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet.

8. The system of claim 7 wherein the connection server, responsive to a positive determination of the on-line status of the second processing unit, retrieves the respective IP address of the second processing unit from the database and transmits the on-line message, including the IP address, to the first processing unit; and

wherein the first processing unit establishes the point-to-point communication link between the first and second processing units through the Internet in response to receiving the IP address of the second processing unit from the connection server.

9. The system of claim 7 wherein the connection server, responsive to a negative determination of the on-line status of the second processing unit, generates an off-line message, and transmits the off-line message to the first processing unit.

10. The system of claim 7 wherein the connection server further includes a timer for timestamping IP addresses of the set of processing units having a positive on-line status with respect to the Internet.

11. The system of claim 7 further comprising:

a mail server for processing a E-mail signal through the Internet to deliver the E-mail to a specified second processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet; and

wherein the first processor of the first processing unit executes a second program to generate and transmit the E-mail signal, including a first IP address

~~associated with the first processing unit, to the mail server.~~

~~12. A method for establishing point-to-point Internet communication comprising the steps of:~~

~~(a) transmitting an E-mail signal, including a first Internet Protocol (IP) address, from a first processing unit;~~

~~(b) processing the E-mail signal through the Internet to deliver the E-mail signal to a second processing unit; and~~

~~(c) transmitting a second IP address to the first processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet.~~

~~13. The method of claim 12 further comprising the step of:~~

~~(a1) generating the E-mail signal from the first IP address corresponding to the first processing unit before the step (a) of transmitting the E-mail signal.~~

14. The method of claim 12 further comprising the step of:

(a1) generating the E-mail signal from a session number before the step (a) of transmitting the E-mail signal.

15. The method of claim 12 wherein the step (b) of processing the E-mail signal further comprises the step of:

(b1) processing the E-mail signal using a mail server operatively connected to the second processing unit.

16. The method of claim 12 further comprising the step of:

(b1) generating a connection signal including the second IP address at the second processing unit before the step (c) of transmitting the second IP address to the first processing unit; and

wherein the step (c) of transmitting the second IP address includes the step (c1) of transmitting the connection signal from the second processing unit to the first processing unit.

~~17. An apparatus comprising:~~

~~a first processing unit including:~~

~~a program stored in a memory for performing a point-to-point Internet protocol; and~~

~~a processor for executing the point-to-point Internet protocol program to generate an E-mail signal, including a first Internet Protocol (IP) address, and for transmitting the E-mail signal through the Internet to a second processing unit for establishing a point-to-point communication link to the first processing unit.~~

~~18. The apparatus of claim 17 wherein the processor is adapted to generate the E-mail signal from the first IP address corresponding to the first processing unit.~~

~~19. A system for point-to-point communications over the Internet comprising:~~

~~a first processing unit including:~~

~~a first program for performing a point-to-point Internet protocol; and~~

~~a first processor for executing the first program and for transmitting an E-mail signal, including a first Internet Protocol (IP) address; and~~

~~a mail server for processing the E-mail signal through the Internet to deliver the E-mail to a~~

second processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet.

20. The system of claim 19 further comprising:

the second processing unit including:

a second program for performing the point-to-point Internet protocol; and

a second processor for executing the second program and for receiving the E-mail signal from the mail server and for generating a connection signal, including a second IP address, for establishing the point-to-point communication link to the first processing unit.

Add A2

add B3

6

479-201

A

08 533115

649-2

POINT-TO-POINT INTERNET PROTOCOL

BACKGROUND OF THE INVENTION

1. Field of the Invention

This disclosure relates to network communication protocols, and in particular to a point-to-point protocol for use with the Internet.

2. Description of the Related Art

The increased popularity of on-line services such as AMERICA ONLINE™, COMPUSERVE®, and other services such as Internet gateways have spurred applications to provide multimedia, including video and voice clips, to online users. An example of an online voice clip application is VOICE E-MAIL FOR WINCIM and VOICE E-MAIL FOR AMERICA ONLINE™, available from Bonzi Software, as described in "Simple Utilities Send Voice E-Mail Online", MULTIMEDIA WORLD, VOL. 2, NO. 9, August 1995, p. 52. Using such Voice E-Mail software, a user may create an audio message to be sent to a predetermined E-mail address specified by the user.

B

Generally, devices interfacing to the Internet and other online services may communicate with each other upon establishing respective device addresses. One type of device address is the Internet Protocol (IP) address, which acts as a pointer to the device associated with the IP

2

address. A typical device may have a Serial Line Internet Protocol or Point-to-Point Protocol (SLIP/PPP) account with a permanent IP address for receiving e-mail, voicemail, and the like over the Internet. E-mail and voicemail is generally intended to convey text, audio, etc., with any routing information such as an IP address and routing headers generally being considered an artifact of the communication, or even gibberish to the recipient.

Devices such as a host computer or server of a company may include multiple modems for connection of users to the Internet, with a temporary IP address allocated to each user. For example, the host computer may have a general IP address "XXX.XXX.XXX~~.XXX~~", and each user may be allocated a successive IP address of ~~XXX.XXX.XXX.XXX.10~~ XXX.XXX.XXX.10, ~~XXX.XXX.XXX.XXX.11~~ XXX.XXX.XXX.11, ~~XXX.XXX.XXX.XXX.12~~ XXX.XXX.XXX.12, etc. Such temporary IP addresses may be reassigned or recycled to the users, for example, as each user is successively connected to an outside party. For example, a host computer of a company may support a maximum of 254 IP addresses which are pooled and shared between devices connected to the host computer.

Permanent IP addresses of users and devices accessing the Internet readily support point-to-point communications of voice and video signals over the Internet. For example, realtime video teleconferencing has been implemented using dedicated IP addresses and mechanisms known as reflectors. Due to the dynamic nature of temporary IP addresses of some devices accessing the Internet, point-to-point communications in realtime of voice and video have been generally difficult to attain.

SUMMARY OF THE INVENTION

A point-to-point Internet protocol is disclosed which exchanges Internet Protocol (IP) addresses between processing units to establish a point-to-point communication link between the processing units through the Internet.

A first point-to-point Internet protocol is disclosed which includes the steps of:

(a) storing in a database a respective IP address of a set of processing units that have an on-line status with respect to the Internet;

(b) transmitting a query from a first processing unit to a connection server to determine the on-line status of a second processing unit; and

(c) retrieving the IP address of the second unit from the database using the connection server, in response to the determination of a positive on-line status of the second processing unit, for establishing a point-to-point communication link between the first and second processing units through the Internet.

A second point-to-point Internet protocol is disclosed, which includes the steps of:

(a) transmitting an E-mail signal, including a first IP address, from a first processing unit;

(b) processing the E-mail signal through the Internet to deliver the E-mail signal to a second processing unit; and

(c) transmitting a second IP address to the first processing unit for establishing a point-to-point communication link between the first and second processing units through the Internet.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the disclosed point-to-point Internet protocol and system will become more readily apparent and may be better understood by referring to the following detailed description of an illustrative embodiment of the present invention, taken in conjunction with the accompanying drawings, where:

FIG. 1 illustrates, in block diagram format, a system for the disclosed point-to-point Internet protocol;

FIG. 2 illustrates, in block diagram format, the system using a secondary point-to-point Internet protocol;

FIG. 3 illustrates, in block diagram format, the system of FIGS. 1-2 with the point-to-point Internet protocol established;

FIG. 4 is another block diagram of the system of FIGS. 1-2 with audio communications being conducted;

FIG. 5 illustrates a display screen for a processing unit;

FIG. 6 illustrates another display screen for a processing unit;

FIG. 7 illustrates a flowchart of the initiation of the point-to-point Internet protocols;

FIG. 8 illustrates a flowchart of the performance of the primary point-to-point Internet protocols; and

FIG. 9 illustrates a flowchart of the performance of the secondary point-to-point Internet protocol.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now in specific detail to the drawings, with like reference numerals identifying similar or identical elements, as shown in FIG. 1, the present disclosure describes a point-to-point Internet protocol and system 10 for using such a protocol.

In an exemplary embodiment, the system 10 includes a first processing unit 12 for sending at least a voice signal from a first user to a second user. The first processing unit 12 includes a processor 14, a memory 16, an input device 18, and an output device 20. The output device 20 includes at least one modem capable of, for example, 14.4 kbaud communications and operatively connected via wired and/or wireless communication connections to the Internet. One skilled in the art would understand that the input device 18 may be implemented at least in part by the modem of the output device 20 to allow input signals from the communication connections to be received. The second processing unit 22 may have a processor, memory, and input and output devices, including at least one modem and associated communication connections, as described above for the first processing unit 12. In an exemplary embodiment, each of the processing units 12, 22 may be a WEBPHONE™

Ins A' unit, available from A' ~~INTERNET TELEPHONE COMPANY,~~ capable of operating the disclosed point-to-point Internet protocol and system 10, as described herein.

The first processing unit 12 and the second processing unit 22 are operatively connected to the Internet 24 by communication devices and software known in the art.

A B The processing units 12, 22 may be ~~are~~ operatively interconnected through the Internet 24 to ~~by~~ a connection server 26, and may also be operatively connected to a mail server 28 associated with the Internet 24.

The connection server 26 includes a processor 30, a timer 32 for generating timestamps, and a memory such as a database 34 for storing, for example, E-mail and Internet Protocol (IP) addresses of logged-in units. In an exemplary embodiment, the connection server 26 may be a SPARC 5 server or a SPARC 20 server, available from SUN MICROSYSTEMS, INC., Mountain View, CA, having a central processing unit (CPU) as processor 30 operating an operating system (OS) such as UNIX and providing timing operations such as maintaining the timer 32, a hard drive or fixed drive as well as dynamic ~~read-only~~ random access memory (DRAM) for storing the database 34, and a keyboard and display and/or other input and output devices (not shown in FIG. 1). The database 34 may be an SQL database available from ORACLE or INFOMIX.

In an exemplary embodiment, the mail server 28 may be a Post Office Protocol (POP) Version 3 mail server including a processor, memory, and stored programs operating in a UNIX environment, or alternatively ~~other~~ another OS, to process E-mail capabilities between processing units and devices over the Internet 24.

The first processing unit 12 may operate the disclosed point-to-point Internet protocol by a computer program described hereinbelow in conjunction with FIG. 6, which ~~are~~ may be implemented from compiled and/or interpreted source code in the C++ programming language and which may be downloaded to the first processing unit 12 from an external computer. The operating computer program may be stored in the memory 16, which may include about 8 MB RAM and/or a hard or fixed drive having about 8 MB. Alternatively, the source code may be implemented in the first processing unit 12 as firmware, as an erasable read only memory (EPROM), etc. It is understood that one skilled in the art would be able to use programming languages other than C++ to implement the disclosed point-to-point Internet protocol and system 10.

The processor 14 receives input commands and data from a first user associated with the first processing unit 12 through the input device 18, which may be an input port connected by a wired, optical, or a wireless connection for electromagnetic transmissions, or alternatively may be transferable storage media, such as floppy disks, magnetic tapes, compact disks, or other storage media including the input data from the first user.

The input device 18 may include a user interface (not shown) having, for example, at least one button actuated by the user to input commands to select from a plurality of operating modes to operate the first processing unit 12. In alternative embodiments, the input device 18 may include a keyboard, a mouse, a touch screen, and/or a data reading device such as a disk drive for receiving the input data from input data files stored in storage media such as a floppy disk or, for example, an 8 mm storage tape. The input device 18 may alternatively include connections to other computer systems to receive the input commands and data therefrom.

The first processing unit 12 may include a visual interface as the output device 20 for use in conjunction with the input device 18 and embodied as one of the screens illustrated by the examples shown in FIGS. 5-6 ~~4-5~~ and discussed below. It is also understood that alternative input devices may be used in conjunction with alternative output devices to receive commands and data from the user, such as keyboards, mouse devices, and graphical user interfaces (GUI) such as WINDOWS™ 3.1 available from MICROSOFT™ Corporation Redmond, WA, executed by the processor 14 using, for example, DOS 5.0. One skilled in the art would understand that other operating systems and GUIs, such as OS/2 and OS/2 WARP, available from IBM CORPORATION, Boca Raton, FL may be used. Other alternative input devices may include microphones and/or telephone handsets for receiving audio voice data and commands, with the first processing unit 12 including speech or voice recognition devices, dual tone multi-frequency (DTMF) based devices, and/or software known in the art to accept voice data and commands and to operate the first processing unit 12.

In addition, either of the first processing unit 12 and the second processing unit 22 may be implemented in a personal digital assistant (PDA) providing modem and E-mail capabilities and Internet access, with the PDA providing the input/output screens for mouse interaction or for touchscreen activation as shown, for example, in FIGS. 4-5, as a combination of the input device 18 and output device 20.

For clarity of explanation, the illustrative embodiment of the disclosed point-to-point Internet protocol and system 10 is presented as having individual functional blocks, which may include functional blocks labelled as "processor" and "processing unit". The functions represented by these blocks may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software. For example, the functions of each of the processors and processing units presented herein may be provided by a shared processor or by a plurality of individual processors. Moreover, the use of the functional blocks with accompanying labels herein is not to be construed to refer exclusively to hardware capable of executing software. Illustrative embodiments may include digital signal processor (DSP) hardware, such as the AT&T DSP16 or DSP32C, read-only memory (ROM) for storing software performing the operations discussed below, and random access memory (RAM) for storing DSP results. Very large scale integration (VLSI) hardware

11

embodiments, as well as custom VLSI circuitry in combination with a general purpose DSP circuit, may also be provided. Any and all of these embodiments may be deemed to fall within the meaning of the labels for the functional blocks as used herein.

The processing units 12, 22 are capable of placing calls and connecting to other processing units connected to the Internet 24, for example, via dialup SLIP/PPP lines. In an exemplary embodiment, each processing unit assigns an unsigned long session number, for example, a 32-bit ~~2^{32}~~ bit long sequence in a *.ini file for each call. Each call may be assigned a successive session number in sequence, which may be used by the respective processing unit to associate the call with one of the SLIP/PPP lines, to associate a <ConnectOK> response signal with a <ConnectRequest> signal, and to allow for multiplexing and demultiplexing of inbound and outbound conversations on conference lines.

For callee (or called) processing units with fixed IP addresses, the caller (or calling) processing unit may open a "socket", i.e. a file handle or address indicating where data is to be sent, and transmit a <Call> command to establish communication with the callee utilizing, for example, datagram services such as Internet Standard network layering as well as transport layering, which may include a

Transport Control Protocol (TCP) or a User Datagram Protocol (UDP) on top of the IP. Typically, a processing unit having a fixed IP address may maintain at least one open socket and a called processing unit waits for a <Call> command to assign the open socket to the incoming signal. If all lines are in use, the callee processing unit sends a BUSY signal or message to the caller processing unit.

As shown in FIG. 1, the disclosed point-to-point Internet protocol and system 10 operate when a callee processing unit does not have a fixed or predetermined IP address. In the exemplary embodiment and without loss of generality, the first processing unit 12 is the caller processing unit and the second processing unit 22 is the called processing unit.

When either of processing units 12, 22 logs on to the Internet via a dial-up connection, the respective unit is provided a dynamically allocated IP address by ~~the~~ a connection service provider ~~server 26~~.

Upon the first user initiating the point-to-point Internet protocol when the first user is logged on to the Internet 24, the first processing unit 12 automatically transmits its associated E-mail address and its dynamically allocated IP address to the connection server 26. The connection server 26 then stores these addresses in the database 34 and timestamps the stored addresses using timer

32. The first user operating the first processing unit 12 is thus established in the database 34 as an active on-line party available for communication using the disclosed point-to-point Internet protocol. Similarly, a second user operating the second processing unit 22, upon connection to the Internet 24 through ~~the~~ a connection service provider ~~server~~ 26, is processed by the connection server 26 to be established in the database 34 as an active on-line party.

The connection server 26 may use the timestamps to update the status of each processing unit; for example, after 2 hours, so that the on-line status information stored in the database 34 is relatively current. Other predetermined time periods, such as a default value of 24 hours, may be configured by a systems operator.

The first user with the first processing unit 12 initiates a call using, for example, a Send command and/or a command to speeddial an NTH stored number, which may be labelled [SND] and [SPD][N], respectively, by the input device 18 and/or the output device 20, such as shown in FIGS. 5-6. In response to either the Send or speeddial commands, the first processing unit 12 retrieves from memory 16 a stored E-mail address of the callee corresponding to the NTH stored number. Alternatively, the first user may directly enter the E-mail address of the callee.

The first processing unit 12 then sends a query, including the E-mail address of the callee, to the connection server 26. The connection server 26 then searches the database 34 to determine whether the callee is logged-in by finding any stored information corresponding to the callee's E-mail address indicating that the callee is active and on-line. If the callee is active and on-line, the connection server 26 then performs the primary point-to-point Internet protocol; i.e. the IP address of the callee is retrieved from the database 34 and sent to the first processing unit 12. The first processing unit 12 may then directly establish the point-to-point Internet communications with the callee using the IP address of the callee.

If the callee is not on-line when the connection server 26 determines the callee's status, the connection server 26 sends an OFF-LINE signal or message to the first processing unit 12. The first processing unit 12 may also display a message such as "Called Party Off-Line" to the first user.

When a user logs off or goes off-line from the Internet 24, the connection server 26 updates the status of the user in the database 34; for example, by removing the user's information, or by flagging the user as being off-line. The connection server 26 may be instructed to update the user's information in the database 34 by an off-line message, such as a data packet, sent automatically from the processing unit of the user prior to being disconnected from the connection server 26. Accordingly, an off-line user is effectively disabled from making and/or receiving point-to-point Internet communications.

As shown in FIGS. 2-4, the disclosed secondary point-to-point Internet protocol may be used as an alternative to the primary point-to-point Internet protocol described above, for example, if the connection server 26 is non-responsive, inoperative, and/or unable to perform the primary point-to-point Internet protocol, as a non-responsive condition. Alternatively, the disclosed secondary point-to-point Internet protocol may be used independent of the primary point-to-point Internet protocol. In the disclosed secondary point-to-point Internet protocol, the first processing unit 12 sends a <ConnectRequest> message via E-mail over the Internet 24 to the mail server 28. The E-mail including the <ConnectRequest> message may have, for example, the subject

[*wp#XXXXXXXX#nnn.nnn.nnn.nnn#emailAddr]

where nnn.nnn.nnn.nnn is the current (i.e. temporary or permanent) IP address of the first user, and XXXXXXXX is a session number, which may be unique and associated with the request of the first user to initiate point-to-point communication with the second user.

As described above, the first processing unit 12 may send the <ConnectRequest> message in response to an unsuccessful attempt to perform the primary point-to-point Internet protocol. Alternatively, the first processing unit 12 may send the <ConnectRequest> message in response to the first user initiating a SEND command or the like.

After the <ConnectRequest> message via E-mail is sent, the first processing unit 12 opens a socket and waits to detect a response from the second processing unit 22. A timeout timer, such as timer 32, may be set by the first processing unit 12, in a manner known in the art, to wait for a predetermined duration to receive a <ConnectOK> signal. The processor 14 of the first processing unit 12 may cause the output device 20 to output a Ring signal to the user, such as an audible ringing sound, about every 3 seconds. For example, the processor 14 may output a *.wav file, which may be labelled RING.WAV, which is processed by the output device 20 to output an audible ringing sound.

The mail server 28 then polls the second processing unit 22, for example, every 3-5 seconds, to deliver the E-mail. Generally, the second processing unit 22 checks the incoming lines, for example, at regular intervals to wait for and to detect incoming E-mail from the mail server 28 through the Internet 24.

Typically, for sending E-mail to users having associated processing units operatively connected to a host computer or server operating an Internet gateway, E-Mail for a specific user may be sent over the Internet 24 and directed to the permanent IP address or the SLIP/PPP account designation of the host computer, which then assigns a temporary IP address to the processing unit of the specified user for properly routing the E-mail. The E-mail signal may include a name or other designation such as a username which identifies the specific user regardless of the processing unit assigned to the user; that is, the host computer may track and store the specific device where a specific user is assigned or logged on, independent of the IP address system, and so the host computer may switch the E-mail signal to the device of the specific user. At that time, a temporary IP address may be generated or assigned to the specific user and device.

Upon detecting and/or receiving the incoming E-mail signal from the first processing unit 12, the second processing unit 22 may assign or may be assigned a temporary IP address. Therefore, the delivery of the E-mail through the Internet 24 provides the second processing unit 22 with a session number as well as IP addresses of both the first processing unit 12 and the second processing unit 22.

Point-to-point communication may then be established by the processing units 12, 22. For example, the second processing unit 22 may process the E-mail signal to extract the <ConnectRequest> message, including the IP address of the first processing unit 12 and the session number. The second processing unit 22 may then open a socket and generate a <ConnectOK> response signal, which includes the temporary IP address of the second processing unit 22 as well as the session number.

The second processing unit 22 sends the <ConnectOK> signal directly over the Internet 24 to the IP address of the first processing unit 12 without processing by the mail server 28, and a timeout timer of the second processing unit 22 may be set to wait and detect a <Call> signal expected from the first processing unit 12.

Realtime point-to-point communication of audio signals over the Internet 24, as well as video and voicemail, may thus be established and supported without requiring permanent IP addresses to be assigned to either of the users or processing units 12, 22. For the duration of the realtime point-to-point link, the relative permanence of the current IP addresses of the processing units 12, 22 is sufficient, whether the current IP addresses were permanent (i.e. predetermined or preassigned) or temporary (i.e. assigned upon initiation of the point-to-point communication).

In the exemplary embodiment, a first user operating the first processing unit 12 is not required to be notified by the first processing unit 12 that an E-mail is being generated and sent to establish the point-to-point link with the second user at the second processing unit 22. Similarly, the second user is not required to be notified by the second processing unit 22 that an E-mail has been received and/or a temporary IP address is associated with the second processing unit 22. The processing units 12, 22 may perform the disclosed point to-point Internet protocol automatically upon initiation of the point-to-point ~~point-to-point~~ communication command by the first user without displaying the E-mail interactions to either user. Accordingly, the disclosed point-to-point Internet protocol may be transparent to the users. Alternatively, either of the first and second users may receive, for example, a brief message of "CONNECTION IN PROGRESS" or the like on a display of the respective output device of the processing units 12, 22.

After the initiation of either the primary or the secondary point-to-point Internet protocols described above in conjunction with FIGS. 1-2, the point-to-point communication link over the Internet 24 may be established as shown in FIGS. 3-4 in a manner known in the art. For example, referring to FIG. 3, upon receiving the <ConnectOK> signal from the second processing unit 22, the first processing unit 12 extracts the IP address of the second processing unit 22 and the session number, and the session number sent from the second processing unit 22 is then checked with the session number originally sent from the first processing unit 12 in the <ConnectRequest> message as E-mail. If the session numbers sent and received by the processing unit 12 match, then the first processing unit 12 sends a <Call> signal directly over the Internet 24 to the second processing unit 22; i.e. using the IP address of the second processing unit 22 provided to the first processing unit 12 in the <ConnectOK> signal.

Upon receiving the <Call> signal, the second processing unit 22 may then begin a ring sequence, for example, by indicating or annunciating to the second user that an incoming call is being received. For example, the word "CALL" may be displayed on the output device of the second processing unit 22. The second user may then activate the second processing unit 22 to receive the incoming call.

Referring to FIG. 4, after the second processing unit 22 receives the incoming call, realtime audio and/or video conversations may be conducted in a manner known in the art between the first and second users through the Internet 24, for example, by compressed digital audio signals. Each of the processing units 12, 22 may also display to each respective user the words "IN USE" to indicate that the point-to-point communication link is established and audio or video signals are being transmitted.

In addition, either user may terminate the point-to-point communication link by, for example, activating a termination command, such as by activating an [END] button or icon on a respective processing unit, causing the respective processing unit to send an <End> signal which causes both processing units to terminate the respective sockets, as well as to perform other cleanup commands and functions known in the art.

FIGS. 5-6 illustrate examples of display screens 36 which may be output by a respective output device of each processing unit 12, 22 of FIGS. 1-4 for providing the disclosed point-to-point Internet protocol and system 10. Such display screens may be displayed on a display of a personal computer (PC) or a PDA in a manner known in the art.

As shown in FIG. 5, a first display screen 36 includes a status area 38 for indicating, for example, a called user by name and/or by IP address or telephone number; a current function such as C2; a current time; a current operating status such as "IN USE", and other control icons such as a down arrow icon 40 for scrolling down a list of parties on a current conference line. The operating status may include such annunciators as "IN USE", "IDLE", "BUSY", "NO ANSWER", "OFFLINE", "CALL", "DIALING", "MESSAGES", and "SPEEDDIAL".

Other areas of the display screen 36 may include activation areas or icons for actuating commands or entering data. For example, the display screen 36 may include a set of icons 42 arranged in columns and rows including digits 0-9 and commands such as END, SND, HLD, etc. For example, the END and SND commands may be initiated as described above, and the HLD icon 44 may be actuated to place a current line on hold. Such icons may also be configured to substantially simulate a telephone handset or a cellular telephone interface to facilitate ease of use, as well as to simulate function keys of a keyboard. For example, icons labelled L1-L4 may be mapped to function keys F1-F4 on standard PC keyboards, and icons C1-C3 may be mapped to perform as combinations of function keys, such as CTRL-F1, CTRL-F2, and CTRL-F3, respectively. In addition, the icons labelled L1-L4 and C1-C3 may include circular regions which may simulate light emitting diodes (LEDs) which indicate that the function or element represented by the respective icon is active or being performed.

Icons L1-L4 may represent each of 4 lines available to the caller, and icons C1-C3 may represent conference calls using at least one line to connect, for example, two or more parties in a conference call. The icons L1-L4 and C1-C3 may indicate the activity of each respective line or conference line. For example, as illustrated in FIG. 5, icons L1-L2 may have lightly shaded or colored circles, such as a green circle, indicating that each of lines 1 and 2 are in use, while icons L3-L4 may have darkly shaded or color circles, such as a red or black circle, indicating that each of lines 3 and 4 are not in use. Similarly, the lightly shaded circle of the icon labelled C2 indicates that the function corresponding to C2 is active, as additionally indicated in the status area 38, while darkly shaded circles of icons labelled C1 and C3 indicate that such corresponding functions are not active.

The icons 42 are used in conjunction with the status area 38. For example, using a mouse for input, a line that is in use as indicated by the lightly colored circle of the icon may be activated to indicate a party's name by clicking a right mouse button for 5 seconds until another mouse click is actuated or the [ESC] key or icon is actuated. Thus, the user may switch between multiple calls in progress on respective lines.

Using the icons as well as an input device such as a mouse, a user may enter the name or alias or IP address, if known, of a party to be called by either manually entering the name, by using the speeddial feature, or by double clicking on an entry in a directory stored in the memory, such as the memory 16 of the first processing unit 12, where the directory entries may be scrolled using the status area 38 and the down arrow icon 40.

Once a called party is listed in the status area 38 as being active on a line, the user may transfer the called party to another line or a conference line by clicking and dragging the status area 38, which is represented by a reduced icon 46. Dragging the reduced icon 46 to any one of line icons L1-L4 transfers the called party in use to the selected line, and dragging the reduced icon 46 to any one of conference line icons C1-C3 adds the called party to the selected conference call.

Other features may be supported, such as icons 48-52, where icon 48 corresponds to, for example, an ALT-X command to exit the communication facility of a processing unit, and icon 50 corresponds to, for example, an ALT-M command to minimize or maximize the display screen 36 by the output device of the processing unit. Icon 52 corresponds to an OPEN command, which may, for example, correspond to pressing the O key on a keyboard, to expand or contract the display screen 36 to represent the opening and closing of a cellular telephone. An "opened" configuration is shown in FIG. 5, and a "closed" configuration is shown in FIG. 6. In the "opened" configuration, additional features such as output volume (VOL) controls, input microphone (MIC) controls, waveform (WAV) sound controls, etc.

The use of display screens such as those shown in FIGS. 5-6 provided flexibility in implementing various features available to the user. It is to be understood that additional features such as those known in the art may be supported by the processing units 12, 22.

Alternatively, it is to be understood that one skilled in the art may implement the processing units 12, 22 to have the features of the display screens in FIGS. 5-6 in hardware; i.e. a wired telephone or wireless cellular telephone may include various keys, LEDs, liquid crystal displays (LCDs), and touchscreen actuators corresponding to the icons and features shown in FIGS. 5-6. In addition, a PC may have the keys of a keyboard and mouse mapped to the icons and features shown in FIGS. 5-6.

Referring to FIG. 7, the disclosed point-to-point Internet protocol and system 10 is initiated at a first processing unit 12 for point-to-point Internet communications by starting the point-to-point Internet protocols in step 54; initiating the primary point-to-point Internet protocol in step 56 by sending a query from the first processing unit 12 to the connection server 26; determining if the connection server 26 is operative to perform the point-to-point Internet protocol in step 58 by receiving, at the first processing unit 12, an on-line status signal from the connection server 26, which may include the IP address of the callee or a "Callee Off-Line" message; performing the primary point-to-point Internet protocol in step 60, which may include receiving, at the first processing unit 12, the IP address of the callee if the callee is active and on-line; and initiating and performing the secondary point-to-point Internet protocol in step 62 if the called party is not active and/or on-line.

Referring to FIG. 8 in conjunction with FIGS. 1 and 3-4, the disclosed point-to-point Internet protocol and system 10 operates using the connection server 26 to perform step 60 in FIG. 7 by starting the point-to-point Internet protocol in step 64; timestamping and storing E-mail and IP addresses of logged-in users and processing units in the database 34 in step 66; receiving a query at the connection server 26 from a first processing unit 12 in step 68 to determine whether a second user or second processing unit 22 is logged-in to the Internet 24, with the second user being specified, for example, by an E-mail address; retrieving the IP address of the specified user from the database 34 in step 70 if the specified user is logged-in to the Internet; and sending the retrieved IP address to the first processing unit in step 72 to establish point-to-point Internet communications with the specified user.

Referring to FIG. 9 in conjunction with FIGS. 2-4, the disclosed secondary point-to-point Internet protocol and system 10 operates at the first processing unit 12 to perform step 62 of FIG. 7. The disclosed secondary point-to-point Internet protocol operates as shown in FIG. 9 by starting the secondary point-to-point Internet protocol in step 74; generating an E-mail signal, including a session number and a first IP address corresponding to a first processing unit in step 76 using the first processing unit 12; transmitting the E-mail signal as a <ConnectRequest> signal to the Internet 24 in step 78; delivering the E-mail signal through the Internet 24 using a mail server 28 to a second processing unit 22 in step 80; extracting the session number and the first IP address from the E-mail signal in step 82; transmitting or sending the session number and a second IP address corresponding to the second processing unit 22 to the first processing unit 12 through the Internet 24 in step 84; verifying the session number received from the second processing unit 22 in step 86; and establishing a point-to-point Internet communication link between the first processing unit 12 and second processing unit 22 using the first and second IP addresses in step 88.

While the disclosed point-to-point Internet protocols and system have been particularly shown and described with reference to the preferred embodiments, it is understood by those skilled in the art that various modifications in form and detail may be made therein without departing from the scope and spirit of the invention. Accordingly, modifications such as those suggested above, but not limited thereto, are to be considered within the scope of the invention.

2317

Cap 2301
K. Sutton
#2
12/14/95

PATENT

Atty. Docket No. 649-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton	Examiner:	
Serial No.:	08/533,115	Group:	Art Unit
Filed:	September 25, 1995	Dated:	October 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL		

Commissioner of Patents
and Trademarks
Washington, D.C. 20231

INFORMATION DISCLOSURE STATEMENT

SIR:

It is respectfully requested that the disclosures discussed below (copies enclosed) and cited in annexed Form PTO-1449 be considered by the Examiner in connection with the above-identified patent application, and that such art be made of record in said application.

No representation is made or intended that a search of the art has been made or that no more relevant disclosures than those listed herein are available.

The items are identified as follows:

U.S. Patent No.	Inventor	Issued
5,150,360	Perlman et al.	Sept. 22, 1992

CERTIFICATE OF MAILING 37 C.F.R. § 1.8(a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Date: October 25, 1995

Anthony Natoli
(Name of person mailing paper)

(Signature of person mailing paper)

U.S. Patent No.	Inventor	Issued
5,204,669	Dorfe et al.	Apr. 20, 1993
5,224,095	Woest et al.	Jun. 29, 1993
5,291,554	Morales	Mar. 1, 1994
5,309,433	Cidon et al.	May 3, 1994
5,321,813	McMillen et al.	Jun. 14, 1994
5,357,571	Banwart	Oct. 18, 1994
5,400,335	Yamada	Mar. 21, 1995

The filing of this information disclosure statement is not an admission that the information cited herein is, or is considered to be, material to patentability as defined in 37 C.F.R. § 1.56(b).

[ ] This Information disclosure statement is being filed concurrently with this application.

[X] This information disclosure statement is being filed within three (3) months of the filing date of this application.

[ ] This information disclosure statement is being filed within three (3) months of the date of entry of the national stage as set forth in 37 C.F.R. § 1.491 in an international application.

[ ] To the best of Applicant(s) knowledge, this information disclosure statement is being filed before the date of mailing of a first Office Action in connection with this case.

[ ] Enclosed herewith is a certificate under 37 C.F.R. § 1.97(e).

[ ] Enclosed herewith is a petition under 37 C.F.R. § 1.97(d)(ii).

[ ] Enclosed by check is the petition fee of $130.00. 37 C.F.R. § 1.17(i)(1))

[ ] Please charge the $130.00 petition fee to Deposit Account No. **04-1121**.

[ ] Enclosed by check is the $200.00 fee required by 37 C.F.R. § 1.17(p).

[ ] Please charge the $200.00 fee required by 37 C.F.R. § 1.17(p) to Deposit Account No. **04-1121**.

[**X**] Please charge any deficiency as well as any other fee(s) which may become due under 37 C.F.R. § 1.16 and/or 1.17 at any time during the pendency of this application, or credit any overpayment of such fee(s) to Deposit Account **04-1121**. Also, in the event any extensions of time for responding are required for the pending application(s), please treat this paper as a petition to extend the time as required and charge Deposit Account No. **04-1121** therefor. **TWO (2) COPIES OF THIS SHEET ARE ENCLOSED.**

Respectfully submitted,

Anthony J. Natoli
Reg. No. 36,223
Attorney for Applicant(s)

DILWORTH & BARRESE
333 Earle Ovington Blvd.
Uniondale, NY 11553
(516) 228-8484
AJN/rmb

ATTORNEY DOCKET NO.: 649-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	--
Group:	
Art Unit:	

Assistant Commissioner for Patents
Washington, D.C. 20231

AMENDMENT TRANSMITTAL LETTER

Sir:

Transmitted herewith for filing in the above identified patent application are the following papers:

[X] Preliminary Amendment

The fee is calculated as follows:

		Previously Paid			
Total Claims	53	- 20	= 33	X	$22.00 = 726.00
Independent Claims	12	- 6	= 6	X	$78.00 = 468.00
			TOTAL		$1,194.00

The Commissioner is hereby authorized to charge any other fees under 37 C.F.R.

ATTORNEY DOCKET NO.: 649-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Glenn W. Hutton
Serial No.: 08/533,115
Filed: September 25, 1995
For: POINT-TO-POINT INTERNET PROTOCOL
Examiner: --
Group:
Art Unit:

Assistant Commissioner for Patents
Washington, D.C. 20231

AMENDMENT TRANSMITTAL LETTER

Sir:

Transmitted herewith for filing in the above identified patent application are the following papers:

[X] Preliminary Amendment

The fee is calculated as follows:

			Previously Paid				
Total Claims	53	-	20	= 33	X	$22.00	= 726.00
Independent Claims	12	-	6	= 6	X	$78.00	= 468.00
				TOTAL			$1,194.00

The Commissioner is hereby authorized to charge any other fees under 37 C.F.R.

§§1.16 and 1.17 that may be required, or credit any overpayment, to our Deposit Account No. 02-3038.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse, Esq.
Reg. No. 35,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, MA 02108
(617) 367-4600

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on 4-5-96

Lorraine McConnell

Lorraine McConnell

April 5, 1996

726-103
468-102

2317
4-26-96
#3/A

ATTORNEY DOCKET NO.: 649-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Glenn W. Hutton
Serial No.: 08/533,115
Filed: September 25, 1995
For: POINT-TO-POINT INTERNET PROTOCOL
Examiner: --
Group:
Art Unit:

CERTIFICATE OF MAILING

I hereby certify that he following Amendment is being deposited with the United States Postal Service as first class mail in an envelope addressed to Assistant Commissioner of Patents, Washington, D.C. 20231 on April 5, 1996.

Lorraine McConnell
Lorraine McConnell

Assistant Commissioner for Patents
Washington, D.C. 20231

PRELIMINARY AMENDMENT

In the Specification

A'

Page 6, line 11, change "Internet Telephone Company" to --NetSpeak Corporation, Boca Raton, FL,--;

Page 6, line 17, change "are" to --may be--;

Page 7, line 2, after "Inc.," insert --Mountain View, CA,--;

Page 7, line 20, change "are" to --may be--.

Page 9, line 6, change "4-5" to --5-6--;

Page 9, line 12, after "corporation" insert --Redmond, WA,--;

Page 9, line 15, after "CORPORATION," insert --Boca Raton, FL,--

31

Page 19, line 14, change "point to-point" to --point-to-point--;

In the Claims

Please add the following claims.

Sub C1

21. A computer program product for use with a computer system, the computer system having first and second processors and a server operatively coupled over a computer network, the computer program product comprising:

a computer usable medium having program code means embodied in the medium for establishing a point-to-point communications link between the first processor and the second processor over the computer network, the medium further comprising:

program code means for transmitting, from the first processor to the server, a query as to whether the second processor is connected to the computer network;

program code means for receiving a network protocol address of the second processor from the server, when the second processor is connected to the computer network; and

program code means, responsive to the network protocol address of the second processor, for establishing a point-to-point communication link between the first processor and the second processor over the computer network.

Sub B1

22. A computer program product for use with a computer system, the computer system having first and second processors and a server operatively coupled over a computer network, the computer program product comprising:

a computer useable medium having program code means embodied in the medium for establishing a point-to-point communications link between the first processor and a second processor over a computer network, the medium further comprising:

~~program code means for transmitting an E-mail signal comprising a network protocol address from the first processor to the server over the computer network;~~

~~program code means for receiving a second network protocol address from the second processor over the computer network; and~~

~~program code means, responsive to the second network protocol address, for establishing a point-to-point communication link between the first processor and the second processor over a computer network.~~

Sub C2

~~23. A computer server apparatus for enabling point-to-point communications between a first and a second processor over a computer network, the server apparatus comprising:~~

~~a server processor;~~

~~a network interface means, operatively coupled to the server processor, for connecting the server apparatus to the computer network;~~

~~a memory, operatively coupled to the processor, for storing a network protocol address for a plurality of processors connected to the computer network;~~

~~means, responsive to a query from the first processor, for determining the on-line status of the second processor and for transmitting the a network protocol address of the second processor to the first processor in response to a positive determination of the on-line status of the second processor.~~

~~24. The computer server apparatus of claim 23 further comprising a timer means, operatively coupled to the server processor, for time stamping the network protocol addresses stored in the memory.~~

~~25. The computer server apparatus of claim 23 further comprising:~~

~~mail processing means, responsive to an E-mail signal from the first processor, for forwarding the E-mail signal to the second processor, the E-mail signal comprising the network protocol address of the first processor.~~

sub C3

A2

~~26. In a connection server having a database and a computer network operatively coupled thereto, a method for enabling point-to-point communication between a first processing unit and a second processing unit over a computer network, the method comprising the steps of:~~

~~A. storing in the database, a respective network protocol address for each of a plurality of processing units that have an on-line status with respect to the computer network;~~

~~B. receiving a query from the first processing unit to determine the on-line status of the second processing unit;~~

~~C. determining the on-line status of the second processing unit; and~~

~~D. transmitting an indication of the on-line status of the second processing unit to the first processing unit over the computer network.~~

~~27. The method of claim 26 wherein step C further comprises the steps of:~~

~~c.1 searching the database for an entry relating the second processing unit; and~~

~~c.2 retrieving the network protocol address of the second processing unit in response to a positive determination of the on-line status of the second processing unit.~~

~~28. The method of claim 26 wherein step D further comprises the steps of:~~

~~d.1 transmitting the network protocol address of the second processing unit to the first processing unit when the second processing unit is determined in~~

step C to have a positive on-line status with respect to the computer network.

29. The method of claim 26 wherein step D further comprises the steps of:

d.1 generating an off-line message when the second processing unit is determined in step C to have a negative on-line status with respect to the computer network; and

d.2 transmitting the off-line message to the first processing unit.

30. The method of claim 26 further comprising the steps of:

E. receiving an E-mail signal comprising a first network protocol address from the first processing unit; and

F. transmitting the E-mail signal over the computer network to the second processing unit.

31. The method of claim 30 wherein the E-mail signal further comprises a session number and wherein step F further comprises the step of:

f.1 transmitting the session number and network protocol address over the computer network to the second processor.

32. A method for establishing a point-to-point communication link from a caller processor to a callee processor over a computer network, the caller processor having a user interface and being operatively coupled to the callee processor and a server over the computer network, the method comprising the steps of:

A. generating an element representing a first communication line;

B. generating an element representing a first callee processor;

C. establishing a point-to-point communication link from the caller processor to the first callee processor, in response to a user associating the element representing the first callee processor with the element representing the first communication line.

33. The method of claim 32 wherein step C further comprises the steps of:

c.1 querying the server as to the on-line status of the first callee processor; and

c.2 receiving a network protocol address of the first callee processor over the computer network from the server.

34. The method of claim 32 further comprising the step of:

D. generating an element representing a second communication line.

35. The method of claim 34 further comprising the step of:

E. terminating the point-to-point communication link from the caller processor to the first callee processor, in response to the user disassociating the element representing the first callee processor from the element representing the first communication line; and

F. establishing a different point-to-point communication link from the caller processor to the first callee processor, in response to the user associating the element representing the first callee processor with the element representing the second communication line.

36. The method of claim 32 further comprising the steps of:

D. generating an element representing a second callee processor; and

E. establishing a conference point-to-point communication link between the caller processor and the first and second callee processors, in response to the user associating the element representing the second callee processor with the element representing the first communication line.

37. The method of claim 32 further comprising the step of:

F. removing the second callee processor from the conference point-to-point communication link in response to the user disassociating the element representing the second callee processor from the element representing the first communication line.

38. The method of claim 32 further comprising the steps of:

D. generating an element representing a communication line having a temporarily disabled status; and

E. temporarily disabling a point-to-point communication link between the caller processor and the first callee processor, in response to the user associating the element representing the first callee processor with the element representing the communication line having a temporarily disabled status.

39. The method of claim 38 wherein the element generated in step D represents a communication line on hold status.

40. The method of claim 39 wherein the element generated in step D represents a communication line on mute status.

41. The method of claim 32 wherein the caller processor further comprises a visual display and the user interface comprises a graphic user interface.

Sub B2

42. The method of claim 41 wherein the elements generated in steps A and B are graphic elements and the step of establishing a point-to-communication link as described in step C is performed in response to a user manipulating the graphic elements on the graphic user interface.

Sub C5

43. A computer program product comprising:

a computer usable medium having program code means embodied in the medium for establishing a point-to-point communication link from a caller processor to a callee processor over a computer network, the caller processor having a user interface and being operatively coupled to the callee processor and a server over the computer network, the medium further comprising:

program code means for generating an element representing a first communication line;

program code means for generating an element representing a first callee processor;

program code means, responsive to a user associating the element representing the first callee processor with the element representing the first communication line, for establishing a point-to-point communication link from the caller processor to the first callee processor.

44. The computer program product of claim 43 wherein the means for establishing a point-to-point communication link further comprises:

program code means for querying the server as to the on-line status of the first callee processor; and

program code means for receiving a network protocol address of the first callee processor over the computer network from the server.

~~45. A computer program product of claim 43 further comprising:~~

~~program code means for generating an element representing a second communication line.~~

~~46. The computer program product of claim 45 further comprising:~~

~~program code means, responsive to the user disassociating the element representing the first callee processor from the element representing the first communication line, for terminating the point-to-point communication link from the caller processor to the first callee processor; and~~

~~program code means, responsive to the user associating the element representing the first callee processor with the element presenting the second communication line, for establishing a different point-to-point communication link from the caller processor to the first callee processor.~~

~~47. The computer program product of claim 43 further comprising:~~

~~program code means for generating an element representing a second callee processor; and~~

~~program code means, responsive to the user associating the element representing the second callee processor with the element representing the first communication line, for establishing a conference communication link between the caller processor and the first and second callee processors.~~

~~48. The computer program product of claim 47 further comprising:~~

~~program code means, responsive to the user disassociating the element representing the second callee processor from the element representing the first communication line, for removing the second callee processor from the conference communication link.~~

49. The computer program product of claim 43 further comprising:

program code means for generating an element representing a communication line having a temporarily disabled status; and

program code means, responsive to user associating the element representing the first callee processor with the element representing the communication line having a temporarily disabled status, for temporarily disabling the point-to-point communication link between the caller processor and the first callee processor.

50. The computer program product of claim 49 wherein the communication line having a temporarily disabled status comprises a communication line on hold status.

51. The computer program product of claim 49 wherein the communication line having a temporarily disabled status comprises a communication line on mute status.

52. A computer program product of claim 43 wherein the caller processor further comprises a visual display and the user interface comprises a graphic user interface.

53. The computer program product of claim 52 wherein the element representing the first communication line and the element representing the first callee processor are graphic elements and wherein the program code means for establishing a point-to-point communication link from the caller processor to the first callee processor further comprises:

program code means, responsive to a user manipulating the graphic elements on the graphic user interface, for establishing the point-to-point communication link from

A2

the caller processor ~~to the first callee~~ processor.

REMARKS

Prior to examining the above-identified application on the merits, Applicant respectfully requests the Examiner to enter the enclosed Preliminary Amendment. Applicant has made minor changes to the specification for greater clarity. No new matter is believed added to the application by the above amendments. In addition, Applicant has added claims 21-53 to more particularly point out and distinctly claim Applicant's inventive contributions to the relevant arts. Support for these claims exists in the specification as filed.

Claims 21-22 and 43-53 conform with In re Beauregard, 35 U.S.P.Q. 2d, 1383 (Fed. Cir. 1995) and the new Patent and Trademark Office policy.

The claims are believed allowable over any of the references cited by the Applicant, whether considered singularly or in combination. Accordingly, Applicant believes this application is in condition for allowance and a notice to that effect is respectfully requested. If the Examiner has any questions regarding this amendment or the application in general he is invited to call the Applicant's attorney at the number listed below.

The Commissioner is hereby authorized to charge any other fees under 37 C.F.R. §1.16 and 1.17 that may be required, or credit any overpayment, to our Deposit Account No. 20-0065.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse
Reg. No. 33,518
Bookstein & Kudirka, P.C.
One Beacon Street

#4
8/9/96

ATTORNEY'S DOCKET NO.: N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT:	Glenn W. Hutton
SERIAL NO.:	08/533,115
FILED:	September 25, 1995
FOR:	POINT-TO-POINT PROTOCOL
EXAMINER:	
ART UNIT:	2305

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231 on July 25, 1996.

Debra M. Doherty — Debra M Doherty
(Typed or printed name of person mailing correspondence) (Signature of person mailing correspondence)

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Transmitted herewith for filing is/are the following document(s):

[XX] Information Disclosure Statement, PTO Form 1449, in duplicate and cited reference

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (617) 367-4600, Boston, Massachusetts.

A check in the amount of $--- is enclosed to cover the filing fee. If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse
Reg. No.:33,518
BOOKSTEIN & KUDIRKA
One Beacon Street
Boston, Massachusetts 02108
Tel.: (617) 367-4600

ATTORNEY DOCKET NO.: N0003/7000
DATE: JULY 25, 1996

ATTORNEY'S DOCKET NO.: N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	
Art Unit:	2305

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Assistant Commissioner for Patents, Washington, DC 20231 on the 25th day of July, 1996.

Debra M Doherty
Debra M. Doherty

Assistant Commissioner for Patents
Washington, DC 20231

STATEMENT FILED PURSUANT TO THE DUTY OF DISCLOSURE UNDER 37 C.F.R. §§1.56, 1.97 AND 1.98

Sir:

Pursuant to the duty of disclosure under 37 C.F.R. §§1.56, 1.97 and 1.98, the applicant requests consideration of this information disclosure statement.

Compliance with 37 C.F.R. §1.97

This information disclosure statement has been filed before the mailing date of a first office action on the merits in the above-identified application. No fee or certification is required.

Information Cited

The applicant hereby makes of record in the above-identified application the information listed on the attached form PTO-1449 (modified). The order of presentation of the references should not be construed as an indication of the relative importance of the references.

Remarks

A copy of each of the above-identified information is enclosed unless otherwise indicated on the attached form PTO-1449 (modified). It is respectfully requested that:

- The examiner consider completely the cited information, along with any other information, in reaching a determination concerning the patentability of the present claims;
- The enclosed form PTO-1449 be signed by the examiner to evidence that the cited information has been fully considered by the Patent and Trademark Office during the examination of this application;
- The citations for the information be printed on any patent which issues from this application.

By submitting this information disclosure statement, the applicant makes no representation that a search has been performed, of the extent of any search performed, or that more relevant information does not exist.

By submitting this information disclosure statement, the applicant makes no representation that the information cited in the statement is, or is considered to be, material to patentability as defined in 37 C.F.R. §1.56(b).

By submitting this information disclosure statement, the applicant makes no representation that the information cited in the statement is, or is considered to be, in fact, prior art as defined by 35 U.S.C. §102.

It is understood by applicant that the foregoing information will be considered and, to the extent deemed appropriate by the examiner, will be reflected in the examiner's communication.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse
Reg. No. 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, Massachusetts 02108
Tel: (617) 367-4600
Attorneys for Applicant

Docket No.: N0003/7000

Date: July 25, 1996

h:\bdj\n0003\7000\\ids.wpd

GP 2305

ATTORNEY'S DOCKET NO.: N0003/7000

#5

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	
Art Unit:	2305

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Assistant Commissioner for Patents, Washington, DC 20231 on the 25th day of August 7, 1996.

Lorraine McConnell

Lorraine McConnell

Assistant Commissioner for Patents
Washington, DC 20231

STATEMENT FILED PURSUANT TO THE DUTY OF <u>DISCLOSURE UNDER 37 C.F.R. §§1.56, 1.97 AND 1.98</u>

Sir:

Pursuant to the duty of disclosure under 37 C.F.R. §§1.56, 1.97 and 1.98, the applicant requests consideration of this information disclosure statement.

Compliance with 37 C.F.R. §1.97

This information disclosure statement has been filed before the mailing date of a first office action on the merits in the above-identified application. No fee or certification is required.

Information Cited

The applicant hereby makes of record in the above-identified application the information listed on the attached form PTO-1449 (modified). The order of presentation of the references should not be construed as an indication of the relative importance of the references.

Remarks

A copy of each of the above-identified information is enclosed unless otherwise indicated on the attached form PTO-1449 (modified). It is respectfully requested that:

- The examiner consider completely the cited information, along with any other information, in reaching a determination concerning the patentability of the present claims;
- The enclosed form PTO-1449 be signed by the examiner to evidence that the cited information has been fully considered by the Patent and Trademark Office during the examination of this application;
- The citations for the information be printed on any patent which issues from this application.

By submitting this information disclosure statement, the applicant makes no representation that a search has been performed, of the extent of any search performed, or that more relevant information does not exist.

By submitting this information disclosure statement, the applicant makes no representation that the information cited in the statement is, or is considered to be, material to patentability as defined in 37 C.F.R. §1.56(b).

By submitting this information disclosure statement, the applicant makes no representation that the information cited in the statement is, or is considered to be, in fact, prior art as defined by 35 U.S.C. §102.

It is understood by applicant that the foregoing information will be considered and, to the extent deemed appropriate by the examiner, will be reflected in the examiner's communication.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse
Reg. No. 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, Massachusetts 02108
Tel: (617) 367-4600
Attorneys for Applicant

Docket No.: N0003/7000

Date: August 7, 1996

h:\bdj\n0003\7000\ids.wpd

ATTORNEY'S DOCKET NO.: N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT:	Glenn W. Hutton
SERIAL NO.:	08/533,115
FILED:	September 25, 1995
FOR:	POINT-TO-POINT INTERNET PROTOCOL
EXAMINER:	
ART UNIT:	2305

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231 on August 7, 1996.

Lorraine McConnell Lorraine McConnell
(Typed or printed name of person mailing correspondence) (Signature of person mailing correspondence)

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Transmitted herewith for filing is/are the following document(s):

[XX] Information Disclosure Statement, PTO Form 1449, and cited reference

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (617) 367-4600, Boston, Massachusetts.

A check in the amount of $--- is enclosed to cover the filing fee. If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse
Reg. No.:33,518
BOOKSTEIN & KUDIRKA
One Beacon Street
Boston, Massachusetts 02108
Tel.: (617) 367-4600

ATTORNEY DOCKET NO.: N0003/7000
DATE: August 7, 1996

#6
J-Epps
9/17/96

ATTORNEY'S DOCKET NO.: N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	
Art Unit:	2305

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Assistant Commissioner for Patents, Washington, DC 20231 on the 6th day of September, 1996.

Lorraine McConnell
Lorraine McConnell

Assistant Commissioner for Patents
Washington, DC 20231

STATEMENT FILED PURSUANT TO THE DUTY OF DISCLOSURE UNDER 37 C.F.R. §§1.56, 1.97 AND 1.98

Sir:

Pursuant to the duty of disclosure under 37 C.F.R. §§1.56, 1.97 and 1.98, the applicant requests consideration of this information disclosure statement.

Compliance with 37 C.F.R. §1.97

This information disclosure statement has been filed before the mailing date of a first office action on the merits in the above-identified application. No fee or certification is required.

Information Cited

The applicant hereby makes of record in the above-identified application the information listed on the attached form PTO-1449 (modified). The order of presentation of the references should not be construed as an indication of the relative importance of the references.

Remarks

A copy of each of the above-identified information is enclosed unless otherwise indicated on the attached form PTO-1449 (modified). It is respectfully requested that:

- The examiner consider completely the cited information, along with any other information, in reaching a determination concerning the patentability of the present claims;
- The enclosed form PTO-1449 be signed by the examiner to evidence that the cited information has been fully considered by the Patent and Trademark Office during the examination of this application;
- The citations for the information be printed on any patent which issues from this application.

By submitting this information disclosure statement, the applicant makes no representation that a search has been performed, of the extent of any search performed, or that more relevant information does not exist.

By submitting this information disclosure statement, the applicant makes no representation that the information cited in the statement is, or is considered to be, material to patentability as defined in 37 C.F.R. §1.56(b).

By submitting this information disclosure statement, the applicant makes no representation that the information cited in the statement is, or is considered to be, in fact, prior art as defined by 35 U.S.C. §102.

It is understood by applicant that the foregoing information will be considered and, to the extent deemed appropriate by the examiner, will be reflected in the examiner's communication.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse
Reg. No. 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, Massachusetts 02108
Tel: (617) 367-4600
Attorneys for Applicant

Docket No.: N0003/7000

Date: September 6, 1996

h:\bdj\n0003\7000\\ids.wpd

ATTORNEY'S DOCKET NO.: N0003/7000 #6

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: Glenn W. Hutton
SERIAL NO.: 08/533,115
FILED: September 25, 1995
FOR: POINT-TO-POINT INTERNET PROTOCOL

EXAMINER: --
ART UNIT: 2305

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231 on September 6, 1996.

Lorraine McConnell Lorraine McConnell
(Typed or printed name of person mailing correspondence) (Signature of person mailing correspondence)

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Transmitted herewith for filing is/are the following document(s):

[XX] Information Disclosure Statement, PTO Form 1449, and references cited

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (617) 367-4600, Boston, Massachusetts.

A check in the amount of $-0- is enclosed to cover the filing fee. If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

Bruce D. Jobse
Bruce D. Jobse
Reg. No.:33,518
BOOKSTEIN & KUDIRKA
One Beacon Street
Boston, Massachusetts 02108
Tel.: (617) 367-4600

ATTORNEY DOCKET NO.: N0003/7000
DATE: September 6, 1996

ATTORNEY DOCKET: N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Glenn W. Hutton

Serial No. 08/533,115

Filed: September 25, 1995

For: POINT-TO-POINT INTERNET PROTOCOL

Examiner:

Art Unit:

Assistant Commissioner for Patents
Washington, DC 20231

LETTER

Dear Sir:

The enclosed Revocation and New Power of Attorney was submitted to the United States Patent and Trademark Office on May 28, 1996 along with two patent assignments and accompanying covers sheets. A copy of the transmittal letter and stamped return postcard which accompanied these documents is enclosed. The Revocation and Power of Attorney form was subsequently returned, possibly erroneously, with a Notice of Recordation of one the assignments dated September 3, 1996. Applicant is herewith submitting the Revocation of Power of Attorney form again so that it may be made of record in the above-identified application.

If the Examiner has any questions regarding this communication or the application in general, he is invited to call Applicant's attorney at the number listed below.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse, Esq.
Reg. No. 33,518
Bookstein & Kudirka, P.C.
One Beacon Street
Boston, MA 02108
(617) 367-4600

Attorney Docket No.: N0003/7000

Date: September 17, 1996

ATTORNEY'S DOCKET NO.:N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: Glenn W. Hutton
SERIAL NO.: 08/533,115
FILED: September 25, 1995
FOR: POINT-TO-POINT INTERNET PROTOCOL
EXAMINER: --
ART UNIT: --

COPY

Assistant Commissioner for Patents
Box Assignment
Washington, DC 20231

Sir:

Transmitted herewith for filing is/are the following document(s):

[XX] Two Patent Assignments With Cover Sheets
[XX] Revocation and New Power of Attorney

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (617) 367-4600, Boston, Massachusetts.

A $40.00 check is enclosed for each Assignment to cover the filing fee. If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

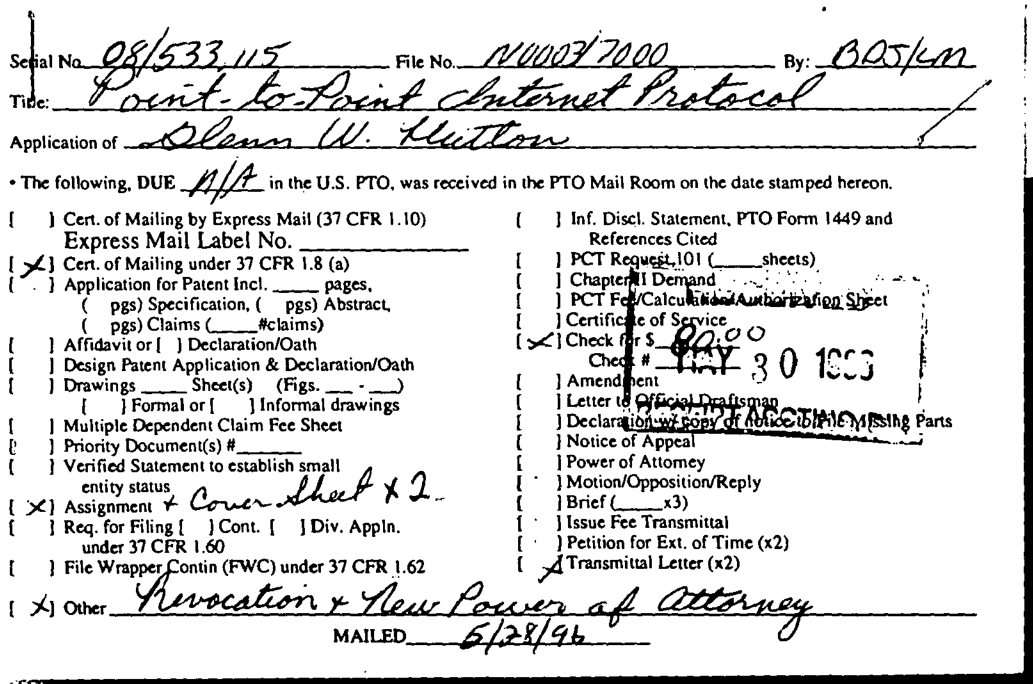
Bruce D. Jobse

Bruce D. Jobse, Esq.
Reg. No.: 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, Massachusetts 02108
Tel.: (617) 367-4600

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on 5/28/96

Bruce D. Jobse

ATTORNEY DOCKET NO.: N0003/7000
DATE: May 28, 1996
XNDD



Serial No. 08/533,115 File No. N0003/7000 By: BDS/km

Title: Point-to-Point Internet Protocol

Application of Glenn W. Hutton

• The following, DUE N/A in the U.S. PTO, was received in the PTO Mail Room on the date stamped hereon.

[ ] Cert. of Mailing by Express Mail (37 CFR 1.10)
Express Mail Label No. ______________
[✓] Cert. of Mailing under 37 CFR 1.8 (a)
[ ] Application for Patent Incl. _____ pages,
(pgs) Specification, (pgs) Abstract,
(pgs) Claims (_____#claims)
[ ] Affidavit or [ ] Declaration/Oath
[ ] Design Patent Application & Declaration/Oath
[ ] Drawings _____ Sheet(s) (Figs. ___ - ___)
[ ] Formal or [ ] Informal drawings
[ ] Multiple Dependent Claim Fee Sheet
[ ] Priority Document(s) #_______
[ ] Verified Statement to establish small entity status
[✓] Assignment + Cover Sheet x 2
[ ] Req. for Filing [ ] Cont. [ ] Div. Appln. under 37 CFR 1.60
[ ] File Wrapper Contin (FWC) under 37 CFR 1.62

[ ] Inf. Discl. Statement, PTO Form 1449 and References Cited
[ ] PCT Request 101 (_____sheets)
[ ] Chapter II Demand
[ ] PCT Fee/Calculation/Authorization Sheet
[ ] Certificate of Service
[✓] Check for $ 80.00
Check # ______
[ ] Amendment
[ ] Letter to Official Draftsman
[ ] Declaration w/ copy of notice to File Missing Parts
[ ] Notice of Appeal
[ ] Power of Attorney
[ ] Motion/Opposition/Reply
[ ] Brief (_____x3)
[ ] Issue Fee Transmittal
[ ] Petition for Ext. of Time (x2)
[✓] Transmittal Letter (x2)

[✓] Other Revocation + New Power of Attorney

MAILED 5/28/96

230

GR235

ATTORNEY'S DOCKET NO.:N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: Glenn W. Hutton
SERIAL NO.: 08/533,115
FILED: September 25, 1995
FOR: POINT-TO-POINT INTERNET PROTOCOL
EXAMINER: --
ART UNIT: --

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Transmitted herewith for filing is/are the following document(s):

[XX] Letter
[XX] Revocation and New Power of Attorney
[XX] Copies of Previous Submission of Transmittal Letter and Stamped Post Card

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (617) 367-4600, Boston, Massachusetts.

If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse, Esq.
Reg. No.: 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, Massachusetts 02108
Tel.: (617) 367-4600

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on 9/17/96

Lorraine McConnell

ATTORNEY DOCKET NO.: N0003/7000
DATE: September 17, 1996
XNDD

PATENT
N003/7000

#7

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Glenn W. Hutton

Serial No. 08/533,115

Filed: September 25, 1995

For: POINT-TO-POINT INTERNET PROTOCOL

Examiner:

Art Unit: 2305

BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, MA 02108

CERTIFICATE OF MAILING

I hereby certify that the following document is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents and Trademarks, Washington, D.C. 20231 on 5/28/96.

Bruce D. Jobse

REVOCATION AND NEW POWER OF ATTORNEY

Netspeak, Corporation, assignee of United States Patent Application Serial No. 08/533,115, filed 9/25/95, hereby revokes all powers of attorney previously given and hereby appoints Arthur Bookstein, Reg. No. 22,958, Paul E. Kudirka, Reg. No. 26,931, Paul J. Cook, Reg. No. 20,820, Bruce D. Jobse, Reg. No. 33,518, Philip L. Conrad, Reg. No. 34,567, Peter M. Dichiara, Reg. No. 38,005, John F. Perullo, Reg. No. 39,498, Christopher S. Daly, Reg. No. 37,303, Steven G. Saunders, Reg. No. 36,265, and BOOKSTEIN & KUDIRKA, P.C. One Beacon Street, Boston, Massachusetts 02108 jointly, and eachof them severally, its attorneys at law, with full power of substitution, delegation and revocation, to prosecute this application to register, to make alterations and amendments therein, to receive the patent, and to transact all business in the Patent and Trademark Office connected therewith. Please direct all telephone calls to Bruce D. Jobse at (617) 367-4600, please address all correspondence to Bruce D. Jobse.

Date: May 9, 1996

By:
Harvey Kaufman
Vice President and Secretary,
NetSpeak, Corp

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
08/533115	09/25/95	HUTTON	649-2

EXAMINER	
PAN, DANIEL	
ART UNIT	PAPER NUMBER
2302	8

JOSEPH J CATANZARO
DILWORTH & BARRESE
333 EARLE OVINGTON BLVD
UNIONDALE NY 11553

DATE MAILED: 10/02/96

10/02/96

This is a communication from
the Patent & Trademark Office

This is in response to the Power of Attorney filed SEPTEMBER 20, 1996.

□ 1. The Power of Attorney to you in this application **has been revoked** by the applicant. Future correspondence will be mailed to the new address of record. 37 CFR 1.33.

☒ 2. The Power of Attorney to you in this application **has been revoked** by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record. (37 CFR 1.33).

□ 3. The withdrawal as attorney in this application **has been accepted.** Future correspondence will be mailed to the new address of record. 37 CFR 1.33.

☒ 4. The Power of Attorney in this application **is accepted.** Correspondence in this application will be mailed to the below-noted address as provided by 37 CFR 1.33.

□ 5. The Power of Attorney in this application **is not accepted** for the reason(s) checked below:

□ a. The Power of Attorney is from an assignee and the Certificate required by 37 CFR 3.73 (b) has not been received.

□ b. The person signing for the assignee has omitted their empowerment to sign on behalf of the assignee.

□ c. The inventor(s) is without authority to appoint attorneys since the assignee has intervened as provided by 37 CFR 3.71.

□ d. The signature of ____________________, a co-inventor in this application, has been omitted. The Power of Attorney will be entered upon receipt of confirmation signed by said co-inventor.

☒ e. The person(s) appointed in the Power of Attorney is not registered to practice before the U.S. Patent & Trademark Office.

□ f. The revocation is not signed by the applicant, the assignee of the entire interest, or one particular principal attorney having the authority to revoke.

BOOSTEIN & KUDIRKA, PC
ONE BEACON STREET
BOSTON, MASSACHUSETTS 02108

Dale A. Hall
DALE A. HALL
GROUP 2300

RETAIN THIS COPY IN THE APPLICATION FILE

#9
11/4/96

PATENT
ATTORNEY DOCKET NO.N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re The Application of: Glenn W. Hutton

Serial No. 08/533,115

Filed: September 25, 1995

For: POINT-TO-POINT INTERNET PROTOCOL

Examiner: D. Pan

Art Unit: 2302

BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, MA 02108
(617) 367-4600

STATUS LETTER

Please inform us of the status for the above-identifed patent application, and when you expect to examine such.

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231, on October 18, 1996

Rosemarie Mullin
Rosemarie Mullin

Date: 10/18/96

Respectfully submitted,

Bruce D. Jobse

Bruce D. Jobse, Esq.
Reg. No. 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, MA 02108
(617) 367-4600

ATTORNEY DOCKET NO. N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	--
Art Unit:	2302

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Assistant Commissioner for Patents, Washington, DC 20231 on the 17th day of April, 1997.

Frances M. Cunningham
Frances M. Cunningham

Assistant Commissioner for Patents
Washington, D.C. 20231

PETITION TO ADD TO ORIGINALLY NAMED INVENTOR(S) UNDER 37 CFR 1.48(c)

Sir:

Applicant respectfully requests that the above-identified application be amended under 37 CFR 1.48(c) to add inventors for subject matter disclosed in the application but previously unclaimed.

Please add the following inventors:

Shane D. Mattaway
826 Periwinkle Street
Boca Raton, FL 33486

Craig B. Strickland
5713 NW 65th Terrace
Tamarac, FL 33321

Attached with this petition are the following:

A. Statement of facts verified by the original-named inventor establishing when the claims to the previously disclosed unclaimed subject matter by the inventors not named in the application were added and the diligence with which this petition and amendment is being made with respect to these facts;

B. Declaration by each of the actual inventors as required under 37 CFR §1.63;

C. Written assent of the assignee; and

D. Payment of the fee required under 37 CFR §1.17(h) of $130.00

A check in the amount of $130.00 is enclosed to cover the filing fee. If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

April 17, 1997

Bruce D. Jobse

Bruce D. Jobse, Esq.
Reg. No. 33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, MA 02108
(617) 367-4600

H:\BDJ\N0003\7000\PETCORR.WPD

Attorney Docket No. N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT Shane D. Mattaway et al.
SERIAL NO.: 08/533,115
FILED: September 25, 1995
FOR: POINT-TO-POINT INTERNET PROTOCOL
EXAMINER:
ART UNIT: 2302

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231, on April 17, 1997.

Frances M. Cunningham
(Typed or printed name of person mailing correspondence)

Frances M. Cunningham
(Signature of person mailing correspondence)

Assistant Commissioner for Patents
Washington, D.C. 20231

ASSENT OF ASSIGNEE

NetSpeak Corporation, the assignee of record for the above-identified U.S. Patent Application, by way of a first assignment dated November 27, 1995 from Glenn W. Hutton to the Internet Telephone Company, Reel 7981, Frame 0020, and a second assignment from the Internet Telephone Company to NetSpeak Corporation dated May 14, 1996, Reel 7981, Frame 0053, hereby consents to the addition of Shane D. Mattaway and Craig B. Strickland as inventors to the application.

4/15/97
Date

Stephen R. Cohen
Chief Executive Officer
NetSpeak Corporation

C:\WINDOWS\TEMP\ASSENTAS.WPD

ATTORNEY DOCKET NO. N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Glenn W. Hutton
Serial No.:	08/533,115
Filed:	September 25, 1995
For:	POINT-TO-POINT INTERNET PROTOCOL
Examiner:	--
Art Unit:	2302

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Assistant Commissioner for Patents, Washington, DC 20231 on April 17, 1997.

Frances M. Cunningham
Frances M. Cunningham

Assistant Commissioner for Patents
Washington, D.C. 20231

STATEMENT OF FACTS IN SUPPORT OF PETITION TO ADD INVENTORS UNDER 37 CFR §1.48(C)

Statement of Facts

1. On September 25, 1995, patent application serial number 08/533,155, entitled "Point-to-Point Internet Protocol" was filed on my behalf, as sole inventor, by Anthony J. Natoli, Esq., Reg. No. 36,223, of the law firm of Dilworth & Barrese, Uniondale, New York, NY.

2. On November 27, 1995 I assigned all right, title and interest in and to the patent application to the Internet Telephone Company, a Florida corporation having a place of business at One South Ocean Boulevard, Suite 305, Boca Raton, Florida 33432.

3. In March of 1996, NetSpeak Corporation, parent corporation of the Internet Telephone Company, retained the services of Bruce D. Jobse, Esq., Reg. No. 33,518, of the law firm of Bookstein & Kudirka, Boston, Massachusetts, to prosecute

the above-identified application.

4. On April 5, 1996 a preliminary amendment to the patent application was filed adding claims 21-53, some of which were directed to subject matter previously disclosed but not yet claimed.

5. I became aware of the preliminary amendment and the additional claims during a telephone conversation with attorney Bruce D. Jobse sometime in late November 1996.

6. On December 11, 1996 I received a copy of the above-mentioned preliminary amendment filed April 5, 1996. I acknowledge that both Shane D. Mattaway and Craig B. Strickland contributed to the subject matter of at least one currently pending claim of the above-identified application. The necessity of naming Shane D. Mattaway and Craig B. Strickland as inventors was discovered sometime between my subsequent review of the copy of the preliminary amendment and the date of this Statement of Facts. A diligent effort has been made to correct this error.

I hereby declare that all statements made herein of my own knowledge are true and that statements made on information and belief are believed to be true and further that the statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both under Section 1001 of Title 18 of United States Code, and that such willful, false statements may jeopardize the validity of the application or any patents issued therefrom.

4-2-97
Date

Glenn W. Hutton
4-2-97 ~~5713 NW 65th Terrace, Tamarac, FL 33321~~
SA 9725 HAMMOCKS BLVD #206
MIAMI, FL. 33196
Citizen: Canada
H:\BDJ\N0003\7000\STMTFACT.WPD

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are stated below next to my name:

I believe I am an original, first and joint inventor the subject matter which is claimed and for which a patent is sought on the invention entitled **POINT-TO-POINT INTERNET PROTOCOL**, the specification of which was filed on September 25, 1995 under Attorney's Docket Number N0003/7000, now U.S. Patent Application Serial No. 08/533,115.

I hereby state that I have reviewed and understand the contents of the above identified patent application, including the claims as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with 37 C.F.R. 1.56.

I hereby claim the benefit of foreign priority under 35 U.S.C. 119 of any foreign application(s) for patent or inventor's certificate having a filing date before that of the application the priority of which is claimed:

Prior Foreign Application(s): Priority Claimed

____________	____________	____________	_____Yes _____No
(Number)	(Country)	(Filing Date)	

I hereby claim the benefit of United States priority under 35 U.S.C. 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in a listed prior United States application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information material to the patentability of this application as defined in 37 C.F.R. 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

____________	____________	____________
(Application Serial #)	(Filing Date)	(Status)
____________	____________	____________
(Application Serial #)	(Filing Date)	(Status)
____________	____________	____________
(Application Serial #)	(Filing Date)	(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Bruce D. Jobse	Reg. No. 33,518	Paul E. Kudirka	Reg. No. 26,931
Arthur Z. Bookstein	Reg. No. 22,958	John F. Perullo	Reg. No. 36,265
Philip L. Conrad	Reg. No. 34,567	Steven G. Saunders	Reg. No. 36,265
Paul J. Cook	Reg. No. 20,280		

Send correspondence to Bruce D. Jobse, BOOKSTEIN & KUDIRKA, P.C., One Beacon Street, Boston, Massachusetts, 02108.

FULL NAME OF INVENTOR: Glenn W. Hutton

INVENTOR'S SIGNATURE: ____________________ DATE: 4-2-97

RESIDENCE: 9725 Hammocks Boulevard, #206, Miami, FL 33196
CITIZENSHIP: Canada
POST OFFICE ADDRESS: 9725 Hammocks Boulevard, #206, Miami, FL 33196

FULL NAME OF INVENTOR: Shane D. Mattaway

INVENTOR'S SIGNATURE: ____________________ DATE: ____________

RESIDENCE: 826 Periwinkle, Boca Raton, FL 33486
CITIZENSHIP: U.S.A.
POST OFFICE ADDRESS: 826 Periwinkle, Boca Raton, FL 33486

FULL NAME OF INVENTOR: Craig B. Strickland

INVENTOR'S SIGNATURE: ____________________ DATE: ____________

RESIDENCE: 5713 NW 65th Terrace, Tamarac, FL 33321
CITIZENSHIP: Canada
POST OFFICE ADDRESS: 5713 NW 65th Terrace, Tamarac, FL 33321

H:\BDJ\N0003\7000\DECL.WPD

DOCKET NUMBER: N0003/7000

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are stated below next to my name:

I believe I am an original, first and joint inventor the subject matter which is claimed and for which a patent is sought on the invention entitled **POINT-TO-POINT INTERNET PROTOCOL**, the specification of which was filed on September 25, 1995 under Attorney's Docket Number N0003/7000, now U.S. Patent Application Serial No. 08/533,115.

I hereby state that I have reviewed and understand the contents of the above identified patent application, including the claims as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with 37 C.F.R. 1.56.

I hereby claim the benefit of foreign priority under 35 U.S.C. 119 of any foreign application(s) for patent or inventor's certificate having a filing date before that of the application the priority of which is claimed:

Prior Foreign Application(s): Priority Claimed

____________________	________________	________________	_____Yes _____No
(Number)	(Country)	(Filing Date)	

I hereby claim the benefit of United States priority under 35 U.S.C. 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in a listed prior United States application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information material to the patentability of this application as defined in 37 C.F.R. 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

________________	____________________	____________________
(Application Serial #)	(Filing Date)	(Status)
________________	____________________	____________________
(Application Serial #)	(Filing Date)	(Status)
________________	____________________	____________________
(Application Serial #)	(Filing Date)	(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

Bruce D. Jobse	Reg. No. 33,518	Paul E. Kudirka	Reg. No. 26,931
Arthur Z. Bookstein	Reg. No. 22,958	John F. Perullo	Reg. No. 36,265
Philip L. Conrad	Reg. No. 34,567	Steven G. Saunders	Reg. No. 36,265
Paul J. Cook	Reg. No. 20,280		

Send correspondence to Bruce D. Jobse, BOOKSTEIN & KUDIRKA, P.C., One Beacon Street, Boston, Massachusetts, 02108.

FULL NAME OF INVENTOR: Glenn W. Hutton

INVENTOR'S SIGNATURE: ____________________ DATE: __________

RESIDENCE: 9725 Hammocks Boulevard, #206, Miami, FL 33196
CITIZENSHIP: Canada
POST OFFICE ADDRESS: 9725 Hammocks Boulevard, #206, Miami, FL 33196

FULL NAME OF INVENTOR: Shane D. Mattaway

INVENTOR'S SIGNATURE: ____________________ DATE: 1/3/97

RESIDENCE: 826 Periwinkle, Boca Raton, FL 33486
CITIZENSHIP: U.S.A.
POST OFFICE ADDRESS: 826 Periwinkle, Boca Raton, FL 33486

FULL NAME OF INVENTOR: Craig B. Strickland

INVENTOR'S SIGNATURE: ____________________ DATE: 1/3/97

RESIDENCE: 5713 NW 65th Terrace, Tamarac, FL 33321
CITIZENSHIP: Canada
POST OFFICE ADDRESS: 5713 NW 65th Terrace, Tamarac, FL 33321

H:\BDJ\N0003\7000\DECL.WPD

ATTORNEY'S DOCKET NO.: N0003/7000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANTS: Shane D. Mattaway et al.
SERIAL NO.: 08/533,115
FILED: September 25, 1995
FOR: POINT-TO-POINT INTERNET PROTOCOL
EXAMINER:
ART UNIT: 2302

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

The undersigned hereby certifies that this document is being placed in the United States mail with first-class postage attached, addressed to Assistant Commissioner for Patents, Washington, DC 20231 on the 17th day of April, 1997.

Frances M. Cunningham
Frances M. Cunningham

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Transmitted herewith for filing are the following documents:

[X] Declaration and Power of Attorney (2)
[X] Assent of Assignee
[X] Petition to Add to Originally Named Inventor(s)
[X] Statement of Facts

If the enclosed papers are considered incomplete, the Mail Room and/or the Assignment Branch is respectfully requested to contact the undersigned collect at (617) 367-4600, Boston, Massachusetts.

A check in the amount of $130.00 is enclosed for filing of Petition to Add to Originally Named Inventor(s). If the fee is insufficient, the balance may be charged to the account of the undersigned, Deposit Account No. 02-3038. A duplicate of this sheet is enclosed.

Respectfully submitted,

Bruce D. Jobse
Bruce D. Jobse, Esq.
Reg. No.:33,518
BOOKSTEIN & KUDIRKA, P.C.
One Beacon Street
Boston, Massachusetts 02108
Tel.: (617) 367-4600

April 17, 1997

```
=> d his
     (FILE 'USPAT' ENTERED AT 13:03:05 ON 16 MAY 1997)
L1              0 S MENG/XP
L2              0 S (MENG?)/XP
                  E AN/XP
L3             82 S E4-E8
L4           3023 S 395/800/CCLS
L5             36 S L3 NOT L4
L6              1 S L5 AND (BROWSER OR INTERNET OR(WORLD WIDE WEB))
L7            616 S  (BROWSER OR INTERNET OR(WORLD WIDE WEB))
L8              0 S L5 AND (BROWSER AND INTERNET AND (WORLD WIDE WEB))
L9              3 S (BROWSER AND INTERNET AND (WORLD WIDE WEB))
L10             6 S WORLD WIDE WEB AND BROWSER
```

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
08/533,115	09/25/95	HUTTON	G 649-2

B3M1/0602

BOOSTEIN & KUDIRKA, PC
ONE BEACON STREET
BOSTON MA 02108

EXAMINER	
GREGSON, R	

ART UNIT	PAPER NUMBER
2302	10

DATE MAILED: 06/02/97

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary	Application No. 08/533,115	Applicant(s) Hutton	
	Examiner Richard J. Gregson		Group Art Unit 2302

☒ Responsive to communication(s) filed on *25 Sep 1995* .

□ This action is **FINAL.**

□ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire *3* month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claims

☒ Claim(s) *1-53* is/are pending in the application.

Of the above, claim(s) ________ is/are withdrawn from consideration.

□ Claim(s) ________ is/are allowed.

☒ Claim(s) *1-53* is/are rejected.

□ Claim(s) ________ is/are objected to.

□ Claims ________ are subject to restriction or election requirement.

Application Papers

□ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

□ The drawing(s) filed on ________ is/are objected to by the Examiner.

□ The proposed drawing correction, filed on ________ is □ approved □ disapproved.

□ The specification is objected to by the Examiner.

□ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

□ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

□ All □ Some* □ None of the CERTIFIED copies of the priority documents have been

□ received.

□ received in Application No. (Series Code/Serial Number) ________ .

□ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: ________ .

□ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of References Cited, PTO-892

☒ Information Disclosure Statement(s), PTO-1449, Paper No(s). *6*

□ Interview Summary, PTO-413

□ Notice of Draftsperson's Patent Drawing Review, PTO-948

□ Notice of Informal Patent Application, PTO-152

--- SEE OFFICE ACTION ON THE FOLLOWING PAGES ---

Part III DETAILED ACTION

1. Claims 1-53 are presented for examination.

2. A shortened statutory period for response to this action is set to expire three (3) months from the date of mailing of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Information Disclosure Statement

3. In view of the extremely large number of references submitted by the Applicant(s) for consideration of this application, the Applicant(s) are requested to identify any references which have particular significance in the prosecution of this application for further consideration by the Examiner. Applicant(s) should also indicate the specific features, corresponding passages, and figures of such references which are believed to be germane to the invention claimed in the application

Specification

4. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

5. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. § 103 which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.
>
> Subject matter developed by another person, which qualifies as prior art only under subsection (f) or (g) of section 102 of this title, shall not preclude patentability under this section where the subject matter and the claimed invention were, at the time the invention was made, owned by the same person or subject to an obligation of assignment to the same person.

7. Claims 1-4 are rejected under 35 U.S.C. § 103 as being unpatentable over Civanlar, et al, (US 5,581,552) in view of Morgan, et al., (US 5,524,254). The claimed invention found within Claim 1 consists of a method for establishing point-to-point Internet communications comprising (a) storing in a database a set of IP addresses for on-line nodes, (b) transmitting a query from a node to a server to determine the status of a second node, and © retrieving the IP address of the second node from the database in to establish communication between the two nodes. Civanlar, et al., in 2-3, teaches a multimedia server which uses a communication protocol in which the requesting node sends a request for communication with another node through a address server, which contains an address database, to obtain the address and routing information necessary to complete the communication. Civanlar, et al., is silent regarding the database searching to match the address with the destination node. Morgan, et al, in columns in columns 3-4, teaches the look-up procedure into the database which is performed to retrieve the matching address from the database for use in initiating communications over an network.

It would have been obvious to one of ordinary in the art at the time the claimed invention was made to include an database and search/retrieval mechanism to locate the needed network address because such a mechanism permits the database to ~~m~~be modified over time to allow dynamic address assignment thus reducing the need to larger address identifiers and thus the amount of data that needs to be transmitted with each packet of data.

Regarding Claim 2, the claimed invention adds the further limitation to the invention found within Claim 1 that steps of obtaining the on-line status and IP address of the second node include the steps of: (b1) sending a query to a server, (c1) searching the server's database, (c2) determining the on-line status of the second node, (c3) retrieving the IP address of the second node, (c4) and transmitting the IP address of the second node from the server to the requesting node. As was discussed above regarding Claim 1, Morgan, el al., in columns 3-4, teaches the look-up procedure into the database which is performed to retrieve the matching address from the database for use in initiating communications over an network. It would have been obvious to one of ordinary in the art at the time the claimed invention was made to include an database and search/retrieval mechanism to locate the needed network address because such a mechanism permits the database to me modified over time to allow dynamic address assignment thus reducing the need to larger address identifiers and thus the amount of data that needs to be transmitted with each packet of data.

Regarding Claim 3 and 4, the claimed invention in Claim 3 adds the further limitation to the invention found within Claim 2 that the claimed process generate and transmit an error message which is sent to the requesting node when the second node's status is off-line. The claimed invention Claim 4 adds the further limitation to the invention found within Claim 1 that secondary communications protocol is used when a off-line status is found. Morgan, et al., in columns 13-14 teaches the process of handling error condition where the requested second node is not available, that the processing terminates gracefully. Implicit within this operation is the transmittal of appropriate messages to the requesting node of this condition with the initiation of error recovery procedures..

8. Claims 5 and 12-16 are rejected under 35 U.S.C. 103 as being unpatentable over Civanlar, et al, (US 5,581,552) in view of Morgan, et al., (US 5,524,254) as applied to claims 1-5 above, and further in view of December, et al. (The World Wide Web Unleased) . The claimed invention in Claim 5 adds the further limitation to the invention found within Claim 4 that performing the secondary communication protocol includes (d1) transmitting an e-mail signal over Internet from the first node with its IP address, (d2) transmitting the message thru the Internet for delivery at the second node, and (d3) transmitting a second IP address to the first node for establishing the point-to-point communications. The combination of Civanlar, et al., and Morgan, et al. teaches the communications mechanism claimed here in utilizing the address server and its database to initiate communications between the two nodes. Neither of these two references teaches the message transport mechanism which is utilized to transmit the various messages between the various processors on the network. December, et al., on pages 6-9 teaches the various message and data types which are readily transported between two nodes attached to the Internet and that each type of message is a format for which blocks of data are sent between different processors. It would have been obvious to one of ordinary skill in the art at the time the claimed invention was made to utilize Internet e-mail messages as the means to transport various requests between two processors attached to the Internet because it is a well defined and well supported data transport means for moving data between processors across the Internet and that the substitution of e-mail as the transport mechanism

for any other message transport means would be within the ordinary skill of the art as these transport means are equivalent means for moving blocks of data between nodes of the network.

Regarding Claim 12, the claimed invention consists of an independent method claim for establishing point-to point communications comprising transmitting an e-mail signal from the first node via the Internet to the second node, each message containing the appropriate IP address to establish, and using these addresses to establish the point-to-point communication. The claimed invention is a simplified version of the method contained within Claim 1 above with the specification that the messages used to communicate between the first and second nodes be transported using e-mail. The combination of Civanlar, et al., and Morgan, et al. teaches the communications mechanism claimed here in utilizing the address server and its database to initiate communications between the two nodes. Neither of these two references teaches the message transport mechanism which is utilized to transmit the various messages between the various processors on the network. December, et al., on pages 6-9 teaches the various message and data types which are readily transported between two nodes attached to the Internet and that each type of message is a format for which blocks of data are sent between different processors. It would have been obvious to one of ordinary skill in the art at the time the claimed invention was made to utilize Internet e-mail messages as the means to transport various requests between two processors attached to the Internet because it is a well defined and well supported data transport means for moving data between processors across the Internet and that the substitution of e-mail as the transport mechanism for any other message transport means would be within the ordinary skill of the art as these transport means are equivalent means for moving blocks of data between nodes of the network.

Regarding Claim 13 and 14, the claimed invention adds the further limitation to the invention found within Claim 12 that the process of transmitting the appropriate email signal includes the first step of generating the signal to be sent before it is transmitted. Implicit within the teaching of Cinvanlar, et al. Is the step of generating all messages that need top be transmitted to other processors before the message is transmitted using its particular transport means.

Regarding Claim 15, the claimed invention adds the further limitation to the invention found within Claim 12 that processing the e-mail message for delivery thru the Internet consists of the processing the e-mail message using the e-mail server connected to the second processor. Implicit with the teachings of December, et al. is the existence of processes running at both nodes of the Internet that are communicating, which includes the e-mail function, to perform the steps necessary to allow the communication to occur. As such, the transmission of data between two nodes must include the use of a process like a mail server to operate at the receiving end of the communication in order for the communication to be successful. It would have been obvious to one of ordinary skill in the art at the time the claimed invention was made to utilize Internet e-mail messages on regularly supported e-mail servers as the means to transport various requests between two processors attached to the Internet because it is a well defined and well supported data transport means for moving data between processors across the Internet and that the substitution of e-mail as the transport mechanism for any other message transport means would be within the ordinary skill of the art as these transport means are equivalent means for moving blocks of data between nodes of the network.

Regarding Claim 16, the claimed invention adds the further limitation to the invention found within Claim 12 that step of processing the e-mail signal followed by transmitting a second IP address include the steps of generating a connection signal which is transmitted to the first node along with the second IP address. Civanlar, et al., in column 11, teaches the use of a signal to initiate the connection between the two nodes along with the all necessary address information needed by the nodes. December, et al., teaches that the communication of these messages can be accomplished using e-mail over the Internet. It would have been obvious to one of ordinary skill in the art at the time the claimed invention was made to utilize Internet e-mail messages as the means to transport various requests between two processors attached to the Internet because it is a well defined and well supported data transport means for moving data between processors across the Internet and that the substitution of e-mail as the transport mechanism for any other message transport means would be within the ordinary skill of the art as these transport means are equivalent means for moving blocks of data between nodes of the network.

9. Claim 6, which teaches an apparatus claims, fail to teach or define above or beyond Claims 1-5 above and are rejected for the same reasons set forth above in the rejections of Claims 1-5, supra.

10. Claims 7-11, which also teaches a set of apparatus claims, fail to teach or define above or beyond Claims 1-5 above and are rejected for the same reasons set forth above in the rejections of Claims 1-5, supra.

11. Claims 17-18, which teaches a set of apparatus claims, fail to teach or define above or beyond the apparatus found within Claims 12-16 above and are rejected for the same reasons set forth above in the rejections of Claims 12-16, supra.

12. Claims 19-20, which also teaches a set of apparatus claims, fail to teach or define above or beyond the apparatus found within Claims 12-16 above and are rejected for the same reasons set forth above in the rejections of Claims 12-16, supra.

13. Claim 21, which teaches a computer program product claim, fail to teach or define above or beyond Claims 1-5 above and are rejected for the same reasons set forth above in the rejections of Claims 1-5, supra.

14. Claim 22, which teaches a computer program product claim, fail to teach or define above or beyond Claims 12-16 above and are rejected for the same reasons set forth above in the rejections of Claims 12-16, supra.

15. Claims 23-25, which also teaches a set of apparatus claims, fail to teach or define above or beyond Claims 1-5 above and are rejected for the same reasons set forth above in the rejections of Claims 1-5, supra.

16. Claims 26-31, which teaches a set of method claims, fail to teach or define above or beyond the apparatus found within Claims 1-5 above and are rejected for the same reasons set forth above in the rejections of Claims 1-5, supra.

17. Claims 32-42, which also teaches a set of method claims, fail to teach or define above or beyond the apparatus found within Claims 12-16 above and are rejected for the same reasons set forth above in the rejections of Claims 12-16, supra.

18. Claims 43-53, which teaches a set of computer program product claims, fail to teach or define above or beyond the apparatus found within Claims 12-16 above and are rejected for the same reasons set forth above in the rejections of Claims 12-16, supra.

Conclusion

19. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

a. Heylighen teaches the basics of Internet communication and the addressing means used therein.

20. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Richard J. Gregson whose telephone number is (703) 305-4392. The examiner can normally be reached on Monday-Thursday from 8:00 a.m. to 5:30 p.m., as well as on alternate Fridays during these same hours.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alyssa H. Bowler, can be reached on (703) 305-9702. The fax phone number for this Group is (703) 308-5358.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9700.

MARK H. RINEHART
PATENT EXAMINER
GROUP 2300

Richard J. Gregson, Esq.
Patent Examiner
Art Unit 2302

May 22, 1997

DEPARTMENT OF COMMERCE
UNITED STATES OF AMERICA

Notice of References Cited

Application No. 08/533,115	Applicant(s) Hutton			
Examiner Richard J. Gregson		Group Art Unit 2302	Page 1 of 1	

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A	5,581,552	12/96	Civanlar, et al.	370	396
	B	5,524,254	6/96	Morgan, et al.	395	500
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	December & Randall, "The World Wide Web Unleased," Samw Publishing, Indianapolis, IN, 1994, pp. 3-24.	12/94
	V	Heylighen, "WorldWideWeb: a distributed hypermedia paradigm for global networking," IEE/INSPEC Database Updates and Additionss (1960-19950 Doc.# 1374618: Proceedings SHARE Spring Conference, pp. 355-368.	4/94
	W		
	X		

Heylighen, [illegible] "World-Wide Web: a distributed hypermedia paradigm for global networking"; 04/18/94; IEE/INSPEC Database Updates and Additions (1960-95) Doc. # 1374618; Proceedings. SHARE Europe Spring Conference Page: 355-68

World-Wide Web: a distributed hypermedia paradigm for global networking

Francis HEYLIGHEN*

PO, Free University of Brussels, Pleinlaan 2, B-1050 Brussels, Belgium
E-mail: fheyligh@vnet3.vub.ac.be

ABSTRACT. The World-Wide Web (WWW) provides a unified client image to an enormous range and variety of information services on the Internet. WWW use presently undergoes a spectacular growth and seems set to surpass that of all other Internet services. WWW combines extreme power with maximal simplicity and ease of use through its metaphor of distributed hypermedia. A WWW document provides formatted text with possibilities of embedded graphics, sound, animation, and hyperlinks. The hyperlinks when selected provide automatic access to other WWW-documents, files or services (FTP, Gopher, News, Telnet, WAIS, Email...) available anywhere on the Internet. This paper reviews some of the freely available WWW software: browsers, servers, and editors. An elementary introduction is given to the underlying protocols: HTML (HyperText Mark-up Language) for the formatting of hyperdocuments, HTTP (Hypertext Transfer Protocol) for the communication between WWW browsers and servers, and URL (Uniform Resource Locators) as a universal addressing scheme for Internet files and services.

The Internet

The Internet, based on the TCP/IP protocol, can be defined as the "network of networks". It interconnects tens of thousands of computer networks in different parts of the world, allowing them to exchange e-mail, files, and commands. Presently, the number of host computers on the Internet is over 2 million, and the number of users is estimated at 15 to 20 million. It reaches some 140 countries in all continents of the world. The Internet is growing at a rate of about 160% per year. Although there are many computer networks (e.g. company networks) that are as yet not connected to the Internet, there seems to be a general tendency to make Internet access universal. Thus, the Internet is

* Senior Research Assistant NFWO (Belgian National Fund for Scientific Research)

replacing or absorbing older international networks such as BITnet or Usenet, and is rapidly becoming *the* global network.

The recently very fashionable discussions about the "Information Superhighway" cannot avoid referring to the Internet. However, many people still feel that the Internet is not a good model for the Information Infrastructure, as conceived by American vice-president Al Gore, and others. The basic criticism is that typical Internet services are based on arcane interfaces, understandable only to computer specialists. These critics would prefer to found the Information Superhighway on simple, intuitive communication systems that everybody understands, such as the telephone or cable TV. On the other hand, no one who has experienced the power of networked computer services would ever like to go back to these apparently more democratic, but much less powerful, media, like fax or telephone.

There is nothing in principle that prevents Internet services to be as easy to use as a TV set. Historically, the developers of the Internet were UNIX programmers who preferred to solve problems in a "quick and dirty" way, and who did not care much about having to memorize complex command sequences. The most general way of communicating with the Internet is still through a text-only command line interface. Typing in certain commands at ones own computer would open a connection or "log-in" to a different computer in another part of the Internet, where totally different commands might be needed, leading to recurrent confusion. E.g. one of the most typical problems when using the telnet (remote log-in) mechanism is being unable to close the connection: different systems require different commands (e.g. "bye", "quit", "end" , or "Control-Z") to stop the session.

Against this background, the World-Wide Web (WWW) can be defined as an attempt to define an interface to all the different Internet services, which is so user-friendly that people without any knowledge of computers would be able to use it. Moreover, this interface should be universal, independent of the type of computer, operating system, or service requested.

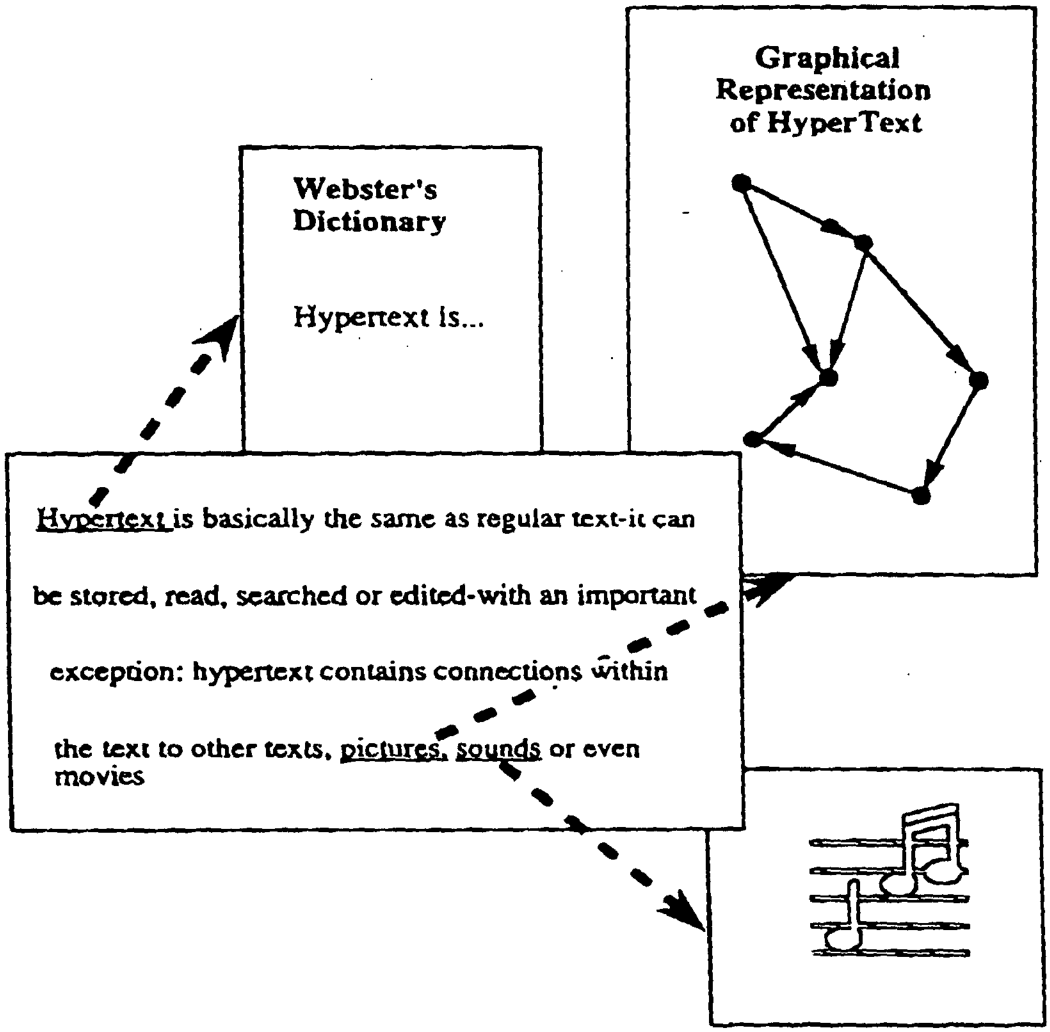
WWW was first conceived in 1989 by Tim Berners-Lee, of CERN (the European Centre for High-Energy Physics in Geneva). Originally proposed as a communication medium for elementary particle physicists, who need to transfer huge amounts of data between countries and continents, it became quickly clear that WWW is useful for all types of networked communication. After a relatively slow start in which the basic protocols were defined, and tested in a few specialised computer centres, 1993 saw a real explosion of WWW use. This was mainly due to the introduction of the popular Mosaic client application by NCSA (the US National Center for Supercomputing Applications). For 1993, the growth rate of WWW-traffic reached an incredible 340,000 %, i.e. magnitudes higher than that of any other Internet service! More and more people seem convinced that WWW is going to overtake all other communication systems on the Internet, thus becoming *the* interface to the global network.

The reason for WWW's popularity lies in its use of an extremely simple but powerful paradigm for representing information available over the network: distributed hypermedia.

The Distributed Hypermedia Paradigm

Hypermedia

Hypertext can be defined as text with "live" references. This means that the text contains highlighted parts (words, phrases, ...), called anchors, which refer to another text. In WWW highlighting is typically done by a blue color, underline, or index number. When the anchor is selected by the user (e.g. by clicking on it with a mouse or by typing in the number), the text referred to is automatically fetched by the computer and shown on the screen. The traditional metaphor is that of an encyclopaedia where related subjects are denoted in the text in bold face, or with a "see also" note. Unlike an encyclopaedia, a hypertext does not require the user to search for the appropriate section by turning over pages: retrieval is done automatically and immediately by the computer (see Fig. 1).



Hypertext is basically the same as regular text-it can be stored, read, searched or edited-with an important exception: hypertext contains connections within the text to other texts, pictures, sounds or even movies

Fig. 1: Hypertext

A hypertext can also be seen as generalization of the typical menu-based or button-based computer interface. Selecting anchors in a text is essentially the same as selecting items from a menu, except that the surrounding, non-active text may provide additional information that can help the user to make the right choice. When the possible choices are presented on the screen there is no longer any need for the user to type in commands, so that the problems of limited memory and incompatible command sets vanishes.

A hypertext referring to another text is said to be "linked" to that text. Links connecting different texts form a directed graph or network, which is called a "web" in WWW terminology. The texts or documents themselves can be viewed as "nodes" in the network or web. Such a web can be extended indefinitely as more and more documents are linked into it.

When the document can contain more than only text, we get an integration of hypertext and multimedia: *hypermedia*. The more sophisticated WWW applications provide formatted text in different sizes, styles or fonts, but also color images, sounds, and even movies or animations. However, the "hyperlinking" is at present limited to text and static images. Links starting from sound or movie files are at present not included in WWW, as their anchors are very complicated to specify.

Images including graphic links are called "maps". When the user clicks on part of the image, the co-ordinates of his or her selection are interpreted by the computer as belonging to a particular region of the image. Depending on the selected anchor region, different linked documents will be fetched. An exemplary application would be a graphical representation or map of a hypertext web, where nodes are represented by rounded rectangles and links by connecting arrows. Clicking on a rectangle would automatically fetch the node document. Another obvious application is a real map of a region where clicking on a particular country will get more detailed information about that country (see Fig. 2).

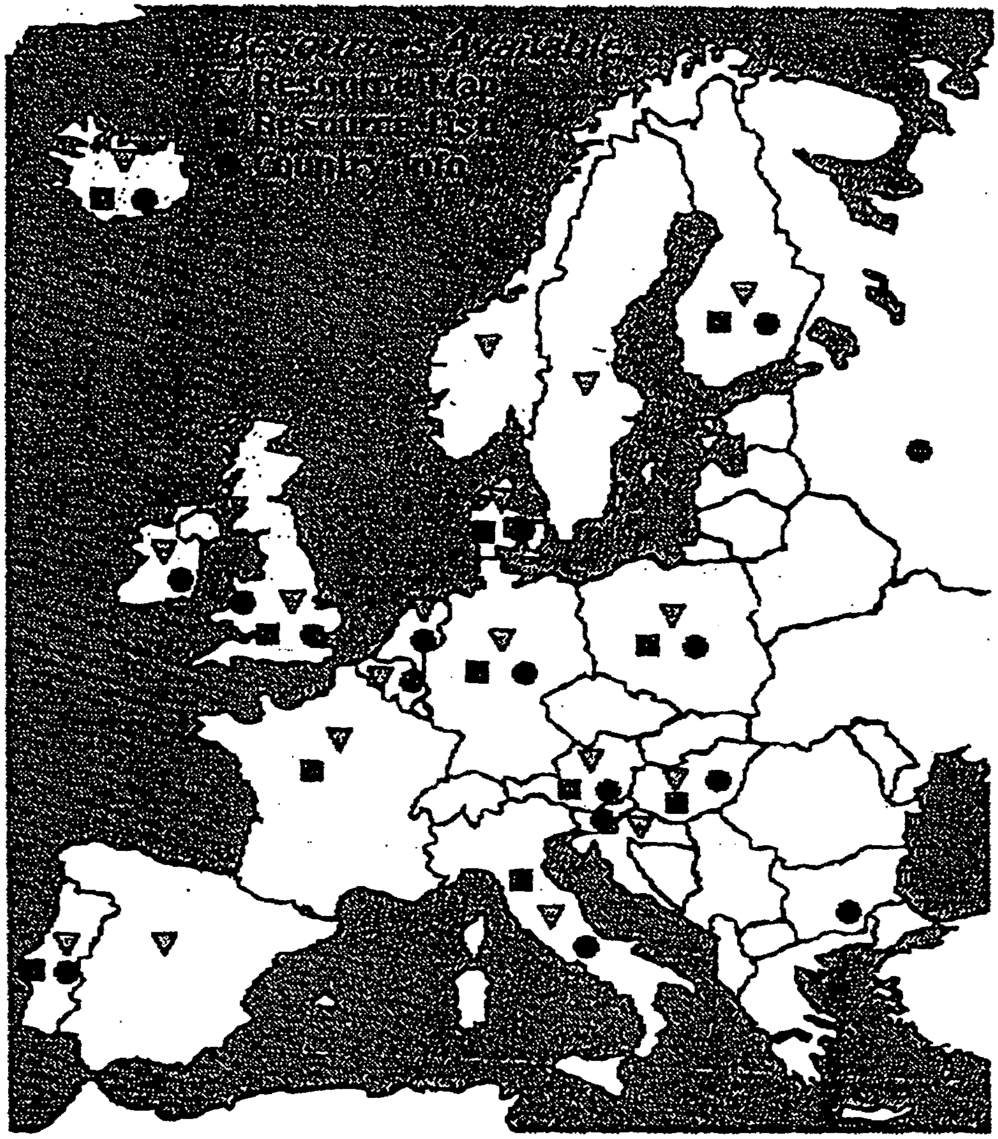


Fig. 2:
Example of a WWW-hypermap: squares, circles and triangles when clicked on will link to other WWW documents in different countries.

Distributed documents

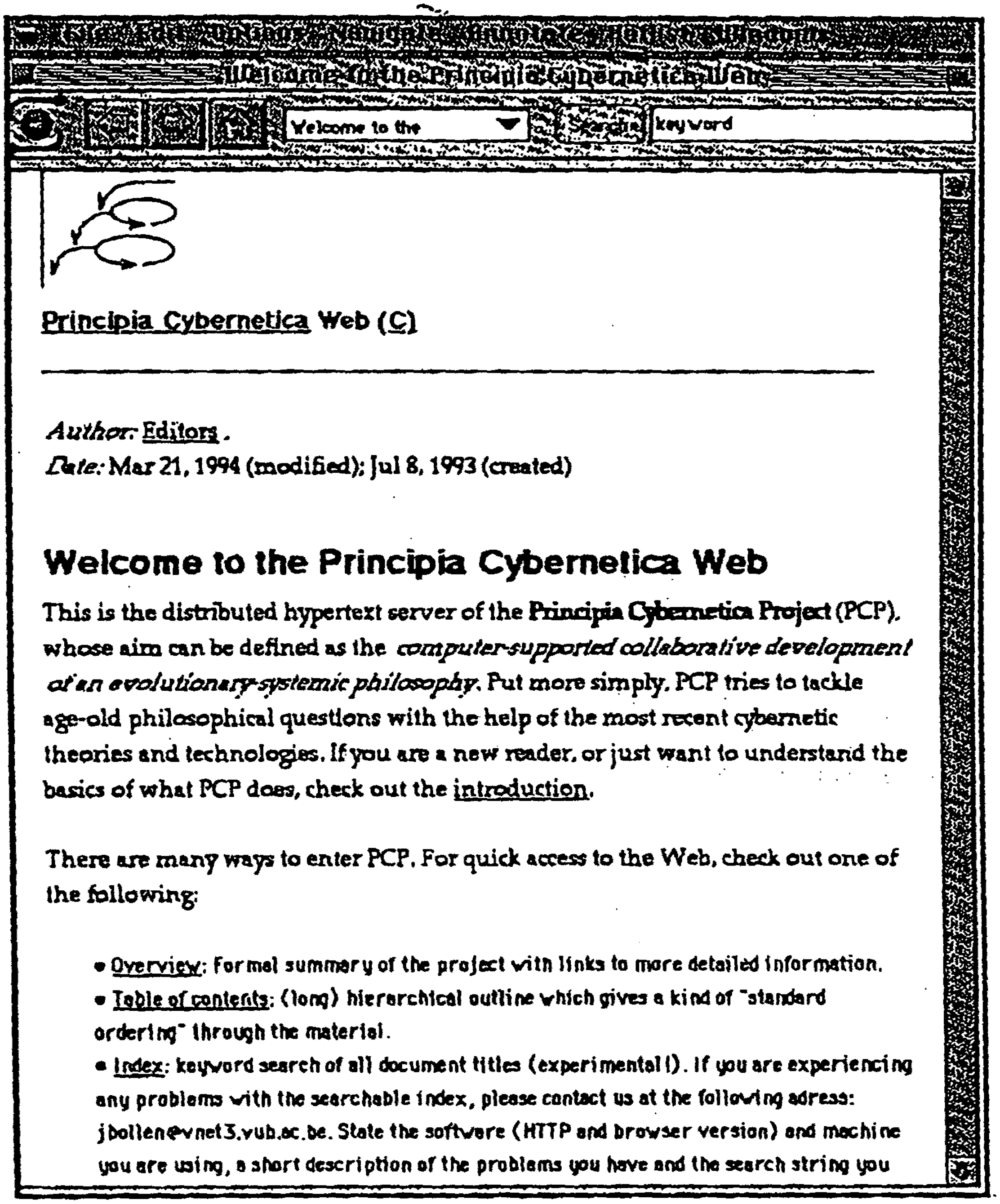
The hypermedia idea was conceived and implemented on different computer systems long before anybody thought of WWW. The originality of Berner-Lee's vision was that he integrated this idea with another idea implicit in the structure of the Internet, that of *distributed information.* The essence of a computer network is that a file no longer needs to be stored on one's local computer in order to be used. With a global network, a document might be stored anywhere in the world, and still its contents could be transferred to your local computer. It then becomes in principle possible to make direct references or links to hyperdocuments residing on remote computers. The above map example (Fig. 2) may then become the representation of geographically distributed information: each dot in a county on the map links to a document not only *about* that country but residing on a computer *in* that country.

The difficulty with the Internet until then was that getting a file from another computer was a quite complicated business. You would first need to know the address of the computer, use that address to open a connection to the remote computer, there locate the file in the local file hierarchy, change the directory, transfer the file to your computer, and finally close the connection. Hypertext, on the other hand, assumes that a simple mouse click should be sufficient to get a document. In order to overcome these obstacles, two things needed to be done: 1) simplify the transfer protocol so that no separate "opening", changing of directories, and "closing" of connections is needed (or at least do these actions automatically in the background so that the user does need to think about them); 2) create a universal addressing scheme which locates not only the remote computer but also its files and their place in the local hierarchy of directories. The first problem led to the definition of the HyperText Transfer Protocol (HTTP), the second to the Uniform Resource Locators (URL).

The net result is that now every document on the Internet can be referred to by a single URL address. When the user selects a link, that address is used to directly contact the server computer and fetch the requested document. The user does not need to know or see the address: the remote document will behave exactly the same as a local document. This makes it possible to geographically distribute information on a given subject, storing it in different places around the globe, and still provide an integrated web of cross-linked documents.

Of course, one remaining difference between local and remote is that it will take more time to transfer a document from the other side of the Earth than from one's own hard disk. In practice, for good Internet lines, and for small documents (a few kilobytes), the difference is hardly noticeable. On the other hand, communications may become frustratingly slow (of the order of minutes to half an hour) when using slow lines (e.g. dial-up modem lines) and transferring large files (e.g. movies). More interestingly, the delay seems to be hardly correlated with physical distance: depending on the quality of the lines, consulting a document on the other side of the ocean may go faster than getting a file from a neighbouring university.

The task of contacting the remote "file server", fetching the file, and visualising it as a hypermedia document is performed by a local application called "browser" (since it browses through pages of information) or "client" (of the server). Moreover, a browser will typically provide a number of "navigation tools", which make it easier for the user to follow paths through the labyrinth of interlinked documents. These include a least a "history" or chronological list of recently consulted documents, and a customizable "hotlist" of documents selected by the user as generally most interesting (see Fig. 3).



Principia Cybernetica Web (C)

Author: Editors.
Date: Mar 21, 1994 (modified); Jul 8, 1993 (created)

Welcome to the Principia Cybernetica Web

This is the distributed hypertext server of the **Principia Cybernetica Project** (PCP), whose aim can be defined as the *computer-supported collaborative development of an evolutionary-systemic philosophy*. Put more simply, PCP tries to tackle age-old philosophical questions with the help of the most recent cybernetic theories and technologies. If you are a new reader, or just want to understand the basics of what PCP does, check out the introduction.

There are many ways to enter PCP. For quick access to the Web, check out one of the following:

- Overview: Formal summary of the project with links to more detailed information.
- Table of contents: (long) hierarchical outline which gives a kind of "standard ordering" through the material.
- Index: keyword search of all document titles (experimental!). If you are experiencing any problems with the searchable index, please contact us at the following adress: jbollen@vnet3.vub.ac.be. State the software (HTTP and browser version) and machine you are using, a short description of the problems you have and the search string you

Fig. 3: Screen shot of the Home Page of a World-Wide Web server as seen through the NCSA Mosaic browser.

Underlined phrases denote hypertext links. The broad arrows in boxes on top of the page allow the user to go backwards or forwards along the sequence of documents visited before. Next to them is a button for the browser's Home Page, and a pop-up menu which list the complete history. Next to it is the search box in which keywords can be entered. The menus on top provide different ways to customize the presentation, navigate the network or create personal hotlists (see text).

Interactive WWW

Until now, all WWW functions we discussed consisted of a passive reading or browsing through pre-existing documents. In most cases, this is all a user interested in getting information needs. However, WWW also provides possibilities for the user to actively enter new information. The simplest and oldest of those is the specification of a search string. Some WWW documents are defined as "indexes", which means that when they are consulted a space appears where the user can type in a set of keywords to be searched for (see Fig. 3). When the search command is given, these keywords are sent to the server, which typically will look up all documents that contain these keywords. in some fashion defined by the programmer. It will produce a list of addresses and titles of these documents, and present them, possibly with additional information (such as file sizes or dates), to the user as a list of hyperlinks. This list is called a "virtual hyperdocument": it looks and behaves like a normal hypertext, but is generated on the fly and was previously not available on the server.

Search boxes only allow entering maximum one line of text. A more recent and more sophisticated WWW application are the "electronic forms". Here the user can fill in different fields (e.g. name, address, age), and choose pre-defined alternatives (e.g. "male" or "female") through pop-up menus or check-boxes. This makes it possible to perform complex database searches, or to enter new records or documents oneself.

Processing of user input by the server does not need to be restricted to search through existing data. One can imagine the most sophisticated programs accepting user-input through forms, and transmitting back their output in the shape of a hypermedia document. A simple existing application is a currency converter, where one types in the amount and specifies the source and target currencies. The program then replies with the amount in the target currency, calculated on the basis of the latest rates. One may envisage much more intelligent programs that can be consulted through a WWW interface. For example, one could imagine a state-of-the-art expert system, continuously updated with the latest medical information, which would accept user problems by letting them fill in forms with symptoms, age, sex, and related data, and respond with advice on possible diseases or treatments in hypermedia format. Another, already existing example illustrates the use of the Web for very specialised scientific domains. Molecular biologists maintain huge databases, available over the Internet, with all known DNA sequences for different organisms. Users can enter sequences they are interested in, and a specialised computer program will then compute all existing sequences that are "similar" to it, in the (non-trivial) sense similarity is determined in genetics, and return the results to the user through a graphical interface.

A most fascinating interactive application of WWW, which is still under development, is the possibility to make "annotations" to existing hyperdocuments. A user who reads a documents, and who would somehow like to make a comment or criticism, can create a new hyperdocument which is linked to the previous one, by an anchor mentioning date, author and subject of the annotation. This is already implemented for "personal annotations", visible only to the user, but is still experimental for "public" and "group" annotations, readable by everybody on the Internet, respectively, by a selected group. This would make it possible for an unlimited amount of people working in different countries to have discussions and collaboratively develop a common web of information.

Integrating Internet Services

The World-Wide Web was designed not only to browse through purpose-built hyperdocuments, but also to use most of the existing Internet services. Since traditional text or image files are just special cases of hyperdocuments without links, this makes it possible to extend the Web of reachable data to a whole universe of pre-existing information. The trick that makes this possible is the previously mentioned URL scheme for addressing different types of resources on the Internet. A URL has the following general form (with some small differences for particular services):

```
method://machine.name/pathname/filename
```

`Method` is an abbreviation of the particular service type, e.g. "FTP", "Gopher", "news" or "Telnet". The `machine name` is the normal Internet host name containing different domains and subdomains separated by dots, e.g. `info.cern.ch` (home of WWW in CERN, Switzerland) or `ftp.ncsa.uiuc.edu` (FTP archive of NCSA in the US). It can be optionally followed by a port number preceded by a colon (e.g. `info.cern.ch:80`). `Pathname` is normally a list of directories and subdirectories separated by slashes (this is more complicated for the Gopher protocol), e.g. `pub/www/bin`. `Filename` is the name of a specific document, often followed by an extension defining the document type, e.g. `default.html`.

Given the information in the URL, a WWW browser should be capable to make the connection using the appropriate protocol. Not all browsers are capable to apply all protocols as yet, but this is just a question of implementation details, not one of lack of specification. We will quickly review the most important services:

FTP: this is most traditional protocol for transferring files to and from computers on the Internet. It is not very user-friendly, as it normally requires the user to open a separate connection, does not provide much information about the contents of a file, and does not allow one to move immediately from one computer to another one.

Gopher: is a more recent protocol for getting files which tackles the above problems. It is similar to WWW in the sense that no opening of connections is needed: it suffices to select a menu item referring to a file on a different computer in order to get that file. The menu gives about one line of information describing the file. A menu item can also link to a different menu on the same or a different computer. This makes "Gopher space", i.e. the set of files and menu's linked in that way resemble a web. However, Gopher fetches either menus of links without additional text, or files without further links. There is thus no real hypertext.

News: the Usenet newsgroups form a system of world-wide "bulletin boards" on different subjects, reachable through many different computers so as to spread the effort of storing and transmitting the huge amount of information they produce. Each day new messages are added and messages older than a predefined period (a few days to a few weeks) are deleted. WWW browsers automatically convert references within a message to other messages or newsgroups into hyperlinks. Since different people consult different host computers, the `//host.name/` part of the URL is not used for newsgroups.

WAIS: is a very powerful and flexible text retrieval system which is accessed by typing in a list of keywords. Different documents available on the server are ordered according to the frequency with which they contain the requested keywords. A list of the documents containing the highest density of keywords is returned to the user.

Telnet: this is the most traditional Internet protocol for logging in to remote computers. It is very inefficient (since a connection with the host computer needs to be kept open during the whole browsing session, including the time spent just for reading the retrieved information), and not user-friendly (since each host computer requires different commands to be typed in, and has minimal capacities for formatting or graphical representation of information). It is still necessary for those Internet services for which no better interface has been developed as yet. The most typical application is the consulting of library databases.

Email: is as yet available on few WWW browsers (as there already exist plenty of good stand-alone email programs). Its URL has the form `mailto://user@host.name`. Its WWW implementation is useful for directly sending a message to the author of a hyperdocument by selecting his or her email address.

Lots of other services are indirectly available on the World-Wide Web through *gateways*: intermediate computers that contact the requested service and automatically convert the exchanged information to the right protocol so that it can be understood by both client and server. Although this is less straightforward than a direct connection, specified by an URL, from the browser to the server, there is no real difference for the user: entering and receiving information happens in exactly the same way. Some of the gatewayed services include : Archie, Veronica (file retrieval), Finger, X-500, WHOIS (finding addresses), HyTelnet, HyperTech (more limited hypertext like services), etc.

How to access WWW

People wishing to use WWW need at least a computer connected to the Internet. This means they need a physical network connection (e.g. a phone line) to the rest of the world, and some implementation of the TCP/IP protocol which defines Internet communication. The connection will normally pass through a name server computer which will assign the user's computer an Internet address

(IP number) and name. The easiest type of connection passes through a local organizational network (e.g. belonging to a company or university), connected via a gateway to the larger Internet. It is also possible to connect via a modem through a telephone line into a gateway computer. This requires a SLIP or PPP connection, which allows the use of TCP/IP over a phone line. Several commercial services are already proposing dial-up Internet connections, although these are often limited to email exchange, lacking the interactive services such as Telnet, FTP and WWW. In these cases it is still possible to consult WWW through a gateway (see further).

Assuming an Internet connection exists, with the possibility of remote log-in, one may try out the Web by using Telnet to one of the following gateways (in mainframe systems, the command is normally "`telnet` " followed by one of the following addresses or IP numbers):

`info.cern.ch`: (IP number `128.141.201.74`)

`ukanaix.cc.ukans.edu` (Full screen browser, requires a vt100 terminal. Log in as www.)

Since these Telnet interfaces are very limited, the next thing to do is to install a browser on the local machine. Browsers exist for almost all computers and operating systems (UNIX, X-Windows, VMS, Windows, Mac, Amiga, ...) and are freely available over the Internet (developed mostly by public institutions such as CERN or NCSA). The software can be retrieved by anonymous FTP at one of the following places:

`ftp.ncsa.uiuc.edu`, in directory `/Mosaic`, `/Mac/Mosaic`, or `/PC/Mosaic`: Mosaic multimedia browser for X-Windows, Mac and MS Windows.

`fatty.law.cornell.edu`, in directory `/pub/LII/cello`: Cello Browser for Microsoft Windows.

`info.cern.ch`, in directory `/pub/www/bin`: Several browsers (Mac, NeXT, DEC...).

For more detailed instructions or for addresses of other browsers, one can consult the Web itself (through Telnet or email) at one of the following URL addresses:

```
http://siva.cshl.org/~boutell/www_faq.html
http://pulua.hcc.hawaii.edu/guide/www.guide.html
http://info.cern.ch/hypertext/WWW/TheProject.html
```

These hyperdocuments contain overviews of how WWW functions and how to get software. Texts on WWW can also be retrieved by anonymous FTP from `info.cern.ch`, directory: `/pub/www/doc`.

People who are not directly connected to the Internet, yet can use email (e.g. through BITnet or CompuServe) can still get WWW-files by sending a message to the email address: `test-list@info.cern.ch` (alternatively `listserv@info.cern.ch`). (this is the CERN mail robot. For more information, send a message containing `HELP` to the same address). The message should consist of one or more lines, each containing the command "`SEND`" followed by the WWW-address (URL) of a desired document. E.g. for getting a list of Frequently Asked-Questions (with answers) on WWW a command line would read:

```
SEND http://siva.cshl.org/~boutell/www_faq.html
```

This will return the hypertext document (text-only), with links numbered. A separate list at the end gives the document-addresses of the linked documents, which can then be requested by a subsequent message. In this way one can navigate through the web, albeit only at mail speed.

Publishing WWW documents

The next step, after consulting WWW documents, is to electronically publish one's own hyperdocuments on the Internet. This is especially interesting for organizations (research centers, companies, associations, ...) that have large amounts of complex information they want to make available to a world-wide public. Thanks to the simple and well-defined protocols, and the freely available server software, this demands much less effort than one might expect. In principle, all documents that are available through FTP, Gopher, WAIS, and similar protocols are simultaneously reachable through World-Wide Web, so it is possible to just use existing Internet channels. However, given the power of the hypermedia format, it is much more interesting to transform existing text files into specific WWW documents.

In order to that, one must know the HyperText Mark-up Language (HTML). This language, developed specifically for WWW but belonging to the SGML (Standard Generalized Mark-Up Languages) family, is quite easy to learn. It uses simple ASCII text to which a number of additional formatting commands are added in the form of "mark-up" tags, distinguished by reserved ASCII symbols such as "<>" brackets. For example, to write a word in bold, one would use the following expression:

```
This is <b>bold</b>.
```

This will be rendered on the screen as: This is **bold**.

Many styles are defined by different mark-up tags: different levels of headers, lists, emphasis, quotes, address, etc. The styles determine basically the structure of the document, not its visual appearance (since this will be different on different platforms).

The most important function of the mark-up is the specification of hyperlinks. This is done basically as in the following example:

```
This is a <a href= "URL">link</a>.
```

`a` stands here for "anchor", `URL` for the address of the linked document. The address can be absolute, including the complete specification of method, hostname, pathname and filename, or relative, including only the part (filename, or pathname/filename) which is different from the address of the present document. Relative addressing not only saves typing effort, but also makes it easier to move a complete set of linked hyperdocuments from one location to another one, without having to change all addresses in the link specifications.

Separate images, sounds or movies can be represented by links as above, where the URL refers to the file to viewed or listened to. An image can be "inlined" in a HTML document with the following mark-up, inserted in that place of the text where the image is supposed to appear. :

```
<img src="image.gif">
```

At present there is still a lack of good (and especially WYSIWYG) editors for HTML. But since the formatting is so simple, everything can be done by hand, and the available editors are expected to become much more sophisticated in the following period.

Assuming that HTML (or plain ASCII) documents have been produced, the next step is to make them available through a server. Since WWW browsers can communicate with other services, this server might be an existing FTP or Gopher server. The protocol used for transferring information from the server to a browser is independent from the mark-up language, so that HTML-formatted files residing in a traditional FTP archive can still be viewed as hyperdocuments. Yet, it is more efficient to use specific WWW-servers implementing the HyperText Transfer Protocol (HTTP). This protocol is simple and stateless, so that a single message from a browser is sufficient to request any file on the server. No connections or directories need to be opened before a file is specified, and this makes HTTP transfers slightly faster than FTP transfers. The most fundamental HTTP command is "`get`" followed by the path name and filename. Through its reliance on simple messages, an HTTP server is similar to an *object*, as in object-oriented programming. Depending on the parameters of the received message, a new message will be generated by the server and sent back to the browser. This makes WWW efficient and modular.

Several servers are freely available on the Internet, for different operating systems. Some of the most used servers are NCSA httpd, CERN httpd, and MacHTTP. They are generally relatively light to install, requiring little memory and processing time. Their Internet locations can be easily found through the different WWW overview documents mentioned earlier.

Conclusion

The World-Wide Web seems to fully deserve the popularity it has gained in a very short time. It provides an extremely powerful interface for the whole of the Internet, and is very simple to use and to install. Presently we are seeing a fast development and elaboration of software (browsers, editors, servers) and underlying protocols (HTML, HTTP), which are becoming more and more effective. Apart from the fact that the Web's popularity will continue to grow very rapidly, few specific predictions can be made about its longer term future. The variety of potential new services (such as public annotations, different forms of interactivity, integration of "virtual reality"-like interfaces) that it may be able to provide a few years from now is so great that envisioning its further development becomes a mind-boggling task. The only thing that can be said at present is that there seems no limit to its extendibility. Of course, it is possible that an even more powerful and efficient set of communication protocols will come to replace WWW, but one may be certain that it will incorporate many of the lessons learned from the present WWW.

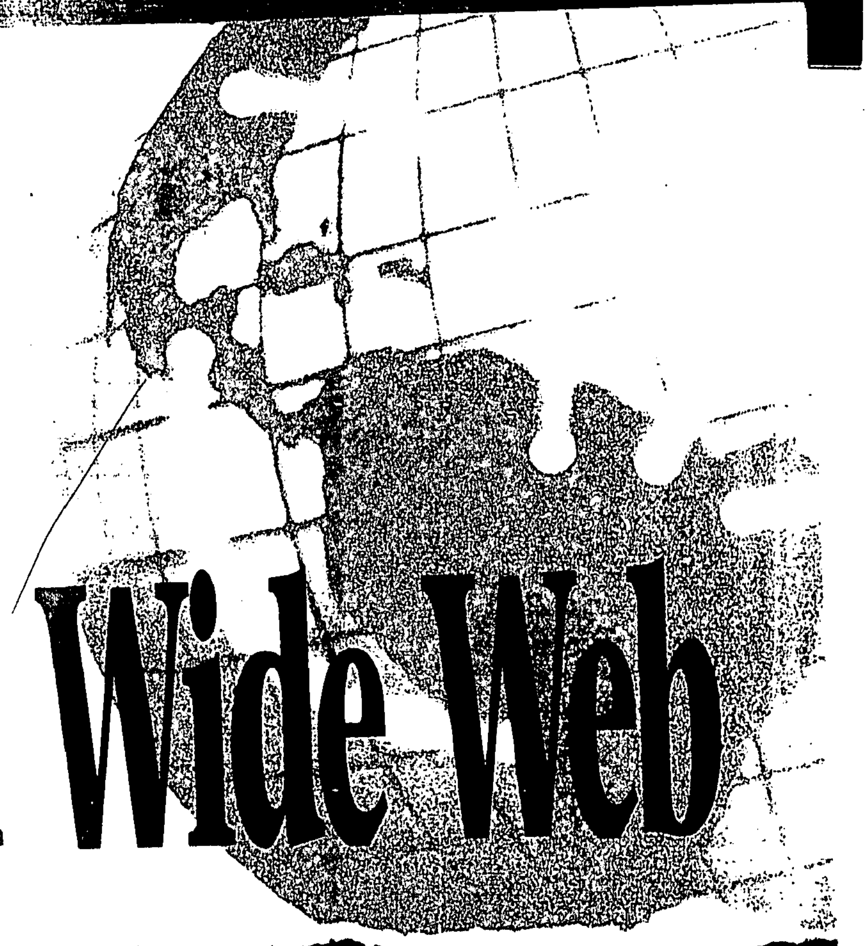
References

Berners-Lee T.J. , R. Cailliau, J-F Groff, B. Pollermann, CERN, "World-Wide Web: The Information Universe", in "*Electronic Networking: Research, Applications and Policy*", Vol. 2 No 1, pp. 52-58 Spring 1992 (Meckler Publishing, Westport, CT, USA).

Berners-Lee T.J., R. Cailliau, N. Pellow, A. Secret, CERN, "The World Wide Web Initiative", *Proceedings of INET93*, San Francisco, 1993.

Berners-Lee, T.J, R. Cailliau and J.-F. Groff (1992), "The World-Wide Web", *Computer Networks and ISDN Systems* 25, p. 454-459.(North-Holland).

Berners-Lee T.J., R. Cailliau, J-F Groff, B. Pollermann (1992), "World-Wide Web: An Information Infrastructure for High-Energy Physics", in: *Proceedings of "Artificial Intelligence and Software Engineering for High Energy Physics"*, edited by D Perret-Gallix (World Scientific, Singapore).



The World Wide Web UNLEASHED

John December
Neil Randall

SAMS PUBLISHING

201 West 103rd Street
Indianapolis, IN 46290

To my family, friends, and everyone on the Net
—John December

To my father, Jacob Lloyd Randall (1918-1994)
—Neil Randall

FIRST EDITION

International Standard Book Number: 0-672-30617-4

Library of Congress Catalog Card Number: 94-67-520

97 96 95 94 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit, the number of the book's printing. For example, a printing code of 94-1 shows that the first printing of the book occurred in 1994.

Composed in MCPdigital by Macmillan Computer Publishing

Printed in the United States of America

Trademarks

ıt topic.
[nternet
d Wide
and are
:s of the
fficult.

ıy a de-
! of the
ɛ Web,

; that's
its full
ges for
ıow, is

1

The World Wide Web: Interface on the Internet

by Neil Randall

IN THIS CHAPTER

For any number of historical reasons, the Internet has emerged as a huge, rich source of information accessible only via a series of not-so-friendly interfaces. The basic commands for telnet, FTP, Archie, WAIS, and even e-mail are powerful but unintuitive, and the rapid growth of the Internet's user base has resulted in an increasing number of users who have neither the patience nor the desire to learn the intricacies of these interfaces.

Even those who know them, however, are aware that easier systems can very quickly result in greater productivity, an awareness that has spawned such eminently usable tools as the popular Gopher. But Gopher is limited as an information source by the restrictions of its display; a gopher is primarily a table of contents through which users read or download files, and tables of contents are useful for some but by no means all types of information reservoirs.

Enter the World Wide Web. Conceptualized not long after Gopher itself, the Web began life as a project designed to distribute scientific information across computer networks in a system known as hypertext. The idea was to allow collaborative researchers to present their research complete with text, graphics, illustrations, and ultimately sound, video, and any other means required.

Important ideas within or across publications would be connected by a series of hypertext links (or just "hyperlinks"), much like the information displays made both possible and plentiful through the Macintosh's famous Hypercard program and similar interfaces available on the NeXT, Amiga, X Window, and Microsoft Windows platforms. Users would be able to traverse Internet documents by selecting highlighted items and thereby moving to other, linked documents, and in the case of graphical displays they would see these documents complete with graphics and other multimedia elements.

The World Wide Web project has made possible the idea of accessible and attractive interfaces on the Internet. Using the Web requires an Internet account and a piece of software known as a World Wide Web client, or browser, and it is the browser's task to display Web documents and allow the selection of hyperlinks by the user.

A few browsers exist that require only text-based displays, the most popular of which is the UNIX program Lynx (now available for DOS machines as well). Most, however, run atop graphical user interfaces such as X Window, the Macintosh, Microsoft Windows, NeXTStep, and Amiga. The most popular browser released to date is Mosaic (available for several of these platforms), but many others are available and in development, both as freeware and as commercially available programs.

With a graphical Web browser, you see formatted documents that contain graphics and highlighted hyperlinks. These browsers let you navigate the Internet not by entering commands, but rather by moving the mouse pointer to the desired hyperlink and clicking. Instantly, the World Wide Web software establishes contact with the remote computer and transfers the requested file to your machine, displaying it in your browser as another formatted, hyperlinked document. You can "surf" the Web by hopping from hyperlink to hyperlink without delving deeply into the contents of any particular document, or you can search the Web for specific documents with specific contents, poring over them as you would a book in the library.

But what *is* the World Wide Web? Where did it come from, and why is it so popular and so potentially important? It is clearly a system of both communication and publication, but how does it work and what can we expect in its future?

These are the questions answered briefly in this chapter and the next four, and examined through a tour in Chapter 6, "The World Wide Web: A Guided Tour." More importantly, however, they're questions explored over the thousand pages of this book, across hundreds of documents on the Web itself, and in magazines, journals, and research reports the world over. The Web is among the most rapidly adopted technological entities of a century that has seen many, and understanding it might be crucial for understanding the next century.

Let's get started.

The Concept of the World Wide Web

The Internet, it is said, is in need of a "killer app." It needs one tool, one program, one application that will take it from being a much-hyped but difficult-to-use linking of computers around the world to being a highly informative, highly usable database and communications tool. The spreadsheet was the killer app for PCs a long time ago, but so far the Net doesn't have one. Some have given "killer app" status to the immensely popular program called Mosaic—see Chapter 10, "The One You Keep Hearing About: NCSA Mosaic," for a lengthy discussion of Mosaic's potential as a killer app—but Mosaic still has its difficulties. The true killer app of the Internet remains somewhere around the corner, and nobody knows if just *one* killer app can handle the Internet's complexity. Until we have one, we simply won't know.

What the Internet does have, however, is a killer *concept*—and the name of that concept is the World Wide Web. In only a few short years of existence, the Web has captured the imagination of data searchers and information surfers alike. Nor is its popularity difficult to understand: The World Wide Web provides the technology needed to offer a navigable, attractive interface for the Internet's vast sea of resources, in much the same way that the toolbar on a word processor screen obscures the intimidating codes that the program actually consists of. Given the Net's history of nearly impenetrable commands and procedures, and the trend in today's software to hide complexity behind usable interfaces, this capability is essential if the Net is become a mainstream set of applications.

But it's important to realize that the Web is a *concept*, not a program, not a system, and not even a specific protocol. It might be more accurate, in fact, to call it an interface but even that wouldn't be quite right. The most accurate terminology might be meta-interface—an interface that incorporates other interfaces—but words with the word "meta" as a prefix went out of favor sometime during the early nineties. Calling it a tool would be far too restrictive, and calling it a set of applications and interfaces would be reasonably accurate but incredibly clumsy. So let's just stick with "concept," because that's as close as we might be able to get.

The Conceptual Make-up of the Web

Calling the Web a *concept*, however, doesn't answer the question of what the World Wide Web actually *is*. Technically, the Web is nothing more than a distributed hypermedia system (at least, that's what its designers call it, as explained in Chapter 5, "Putting It All Together: The World Wide Web"), but *distributed hypermedia system* is surely no more understandable a term than *concept* itself.

The next four chapters examine the variety of systems that constitute the World Wide Web, of which there are, primarily, three:

- The first is *hypertext.*
- The second is *the Internet* itself.
- And the third is that most overused of 1990s terms, *multimedia.*

Important to keep in mind, however, is that the Web is truly a convergence of these systems, in a way that renders the whole much greater than the sum of the parts.

Right now, though, let's concentrate on defining the World Wide Web, or at least providing a definition that helps understand both its past and its future. To do so, we must turn to the three ideas mentioned above: hypertext, the Internet, and multimedia.

Hypertext is an idea that was introduced way back in the seventies by the sometimes visionary, sometimes flaky, and always provocative Ted Nelson. Hypertext is discussed in Chapter 3, "Hypertext," but the idea is deceptively simple. A hypertext document is one that provides clearly visible links to other documents, and in a hypertext computer environment selecting a link in one document moves you directly to the other. Nelson's idea was to link all the world's information in a huge hypertext system, and the World Wide Web is closer than any other system so far to accomplishing that idea, even though it remains a long, long way from fulfilling Nelson's vision.

The second system inherent in the Web's design is the Internet. Covered more comprehensively in Chapter 2, "The Internet," and in fact through a large array of books on the shelves of libraries and bookstores right now, the Internet is a global system of networked computers that allow user-to-user communication and transfer of data files from one machine to any other on the network.

The Net is the basis of the fictional *matrix* or *web* found in the science fiction of such authors as William Gibson and Bruce Sterling, and the basis, as well, of the Clinton administration's much-hyped information superhighway (or, more properly, Global Information Infrastructure). The World Wide Web, in fact, is the closest thing we have now to approximating any of those fictional or semi-fictional technologies.

It's important to note, however, that *the Web as a system does not require the Internet.* In fact, a distributed information system based on the Web can be constructed on *any* local-area or wide-area network, and in fact such systems are being developed all the time.

But the first two words in *World Wide Web* are "world wide," so it makes little sense to talk about the Web without basing it in the world-wide networking—and the only (relatively) open (relatively) world-wide network now available is the Internet. As a result, we'll build the Internet into our definition.

Hypertext, and the Internet. Nice, but not good enough. There's another concept involved as well: *multimedia.*

Again, multimedia is explored more fully in a short while (Chapter 4, "Multimedia," to be exact), but for now let's just say that, as its name suggests, multimedia combines various presentational technologies in an effort to appeal to as many senses as possible. (Actually, the word should be multimedi*um*—like multipart, multisession, multigerm, and multilane—but we'll let the linguists battle over that one.)

Put a bit more simply, multimedia draws on graphics, sound, animation, and video to create a full, rich computing experience. And for the first time, through browsers like Mosaic, Cello, MacWeb, Viola, and others, the World Wide Web offers a multimedia experience for Internet users.

While certainly in need of further development, the Web already lets information presenters place graphics, sound, and video within the page, and users with a direct, high-speed connection can download them quickly enough to feel as if they're participating in full multimedia. With a 14.4 kbps modem the download process is far too slow, but within the next couple of years high-speed access should be much more available and affordable. The important point is that the groundwork has been laid.

So what is it, then? Let's try this: *The World Wide Web is a convergence of computational concepts for presenting and linking information dispersed across the Internet in an easily accessible way.*

Does this help? Well, maybe. Other definitions of the Web tend to use phrases like "network information delivery system" and "distributed information system" and so forth, and no matter how technically accurate these definitions are they just don't seem very useful, because every term with them needs an individual definition as well. Arguably, so does the rather vague *concept* in our own definition, but we know enough about the word *concept* not to need a firm definition. *Concept* is uncertain, volatile, and difficult to grasp, but so is the Web itself—not as a definable computer technology, but rather as a combination of its specifications and its uses. Using the term *concept* might seem like an author's unnecessary avoidance, but anything more precise would almost certainly be outdated within months.

In its initial proposal (discussed in Chapter 5, "Putting It All Together with the Web,") the Web was simply termed "a hypertext project," but it clearly became more than that. What our new definition attempts to do is explain that the Web is a cleverly designed collection of interesting concepts, and allow for the very real possibility that other concepts will soon merge with it.

In fact, this is already happening. Technologies such as WAIS (Wide Area Information Servers) and Archie (the long-lived search engine) are already being programmed into Web-based search tools, and this means that some of the Internet's techniques are already becoming integrated into the Web's conceptual framework. The most successful technologies are those that make its individual components transparent; in the case of the World Wide Web, this seems to be happening early in its history.

The Web contains the technologies necessary to give the Internet a pretty face. Web browsers that take full advantage of these technologies make the Internet easier to use. It's not hard to see where in the history of computing these two crucial ideas—attractiveness and usability—came from. Essentially, the Web and its browsers have done for the Internet in 1994 what the Macintosh did for the personal computer a decade earlier. There were problems with the first Macs from a technological standpoint, and they were written off as toys by the business and computing communities, but they hung on and thrived on the strength of their interface.

Simply put, people could use Macintoshes easily, and that's something that was never true of the IBM PC or its mainframe predecessors. The Mac hid the difficulties of command-line computing under a bunch of objects you could click on with a funny-shaped thing called a mouse, and in doing so it opened computing to the masses. When Microsoft released Windows 3.0 some years later, the iconic, graphic, point-and-click interface (which had originally been developed by Xerox), the masses indeed took over.

Ten years later, graphical World Wide Web browsers such as Mosaic, WinWeb, and MidasWWW offer an interface that has its technological problems, that oversimplifies some important Internet procedures, and that has been called a toy for people who want to glide over the Net rather than delve into it. But just like the Mac, it has thrived because of its interface, and at this time it threatens to overtake all other Internet use, perhaps even the most important Internet tool, electronic mail.

Actually, this comparison between the Mac and the Web isn't quite true, because although the Mac offered just one interface, the Web itself allows all kinds. Its most important interface, however—the graphical, multimedia, point-and-click system offered by Cello and

Mosaic (and others we'll examine in Part II, "Web Browsers and Connections")—is attractive for precisely the same reasons as the Mac and Windows. No matter what its detractors might argue, the World Wide Web offers the Internet to the masses, and that's its true power. No longer do people have to master the vagaries of FTP and Archie and WAIS searching, and as the Web fully develops it should fully incorporate e-mail, newsgroups, telnetting, and other technologies as well.

Different front-ends to the Web will compete for our attention—currently we have Lynx, Viola, Mosaic, Cello, MacWeb, WinWeb, and others—but the principle will remain the same: Link the information, let the users follow whatever path they choose, and once they reach their destination, let them do with the information whatever they please.

It's entirely possible, in fact, that the term "World Wide Web" will become synonymous with the term "Internet," and that's what this book, ***The World Wide Web Unleashed,*** is largely about. If you wish to master the Internet through the mid- to late-nineties, you can't possibly do so without mastering the Web as well.

Even at this early stage in its development, paying attention to the World Wide Web is crucial. It stands poised to become the basis for the revolution in information and connectivity we've all read about but are still waiting to see. You can browse it, search it, and add your own information to the swiftly expanding sea of Web materials. In many ways, it's there for the taking. Already, the Web has begun to change the face of marketing, customer service, business transactions, education, travel, publishing, information dissemination, and collaborative research. What the Web changes in the future is largely up to us. That, so far, is what makes it so fascinating.

The Internet

2

by Neil Randall

IN THIS CHAPTER

Even though the World Wide Web as a system can operate on any computer network, the Web as we know it is nothing without the Internet. In order to understand the Web's importance, in fact, it is necessary to understand the tools, the fundamentals, and even the history of the Net. But a book about the Web is no place for a fully comprehensive look at the Internet; instead, we'll examine the aspects that are especially pertinent for anyone who wants to know about the Web.

A Very Brief History of the Internet

It's been said often, but it bears repeating once more: The Internet was originally conceived by the U.S. military as a means of ensuring a workable communications system in the event of a strike by enemy missiles or forces. It was the sixties, after all, at the height of the Cold War, when the fear of Soviet attack guided all kinds of military projects. If one central communications location was bombed out of existence, the military wanted to make sure that surviving locations could still talk to one another, and that no communication would be lost.

Because the original network was developed by the Advanced Research Projects Agency of the U.S. military, it was given the name ARPAnet. Eventually, however, as increasing numbers of research institutes and research universities connected themselves to the network, ARPAnet came to handle only this kind of research data while a second network, MILnet, looked after military communications. In the 1980s, the National Science Foundation established NSFnet, linking a half-dozen supercomputers at an extremely high speed that has since been made higher still. NSFnet eventually took over the Internet (as it was now being called) from ARPAnet, and in 1991 the U.S. High Performance Computing Act established the basis for the National Research and Education Network (NREN). NREN's goals are to establish and maintain high-speed, high-capacity research and education networks, while helping to develop commercial presence on the Internet as well.

This last point is immensely important for the World Wide Web, which is rapidly being adopted as a medium of choice for businesses in North America and, increasingly, around the globe. During the Internet's early years, commercial activity was severely constrained by the NSF's "Acceptable Use Policy" (AUP), which directly disallowed any for-profit activities. The AUP has changed somewhat, but more importantly the Internet has taken on different forms and different policies. Although it's not actually stated anywhere, commercial activity is now very much accepted on the Net. Whether or not General Motors will begin to sell its vehicles over the World Wide Web remains to be seen, but already the Web is being used for product ordering and product support—and by very sizeable corporations, too.

The Internet has changed so much, in fact, that during the first half of 1994 the number of domain names for commercial organizations (the com domain) overtook those for educational institutions (the edu domain). In the month ending June 25, over 1,300 new commercial (com) names were registered with the Internet, and the following month saw an additional 1,700. That's a 30 percent jump in just one month! And these businesses aren't just moving onto the Net to do research or e-mail, either; they're there because the Internet offers enormous commercial potential.

To be sure, the Internet is still primarily a research and academic network, at least from the standpoints of creative use and extent of use. There's an enormous amount of activity happening in the educational field as well (with the K-12 area burgeoning), and a great deal involving community and nonprofit issues in addition. That's almost certain to change, however, over the second half of the 1990s. The only question now is whether or not governmental legislation will stop the Internet's amazing growth, and the jury's still very much out on that one.

A Very Basic Knowledge of the Internet

To understand the World Wide Web fully, it's essential to know a few significant Internet issues. Actually, the more you know about the Net, the better you know the Web as well, despite the fact that one of the Web's primary functions is to hide users from the difficulties regarding the interfaces of the Internet's tools. The Web is an important layer of functionality and accessibility riding atop the Net, but without the Net and its horde of concepts, the Web would simply be impossible. You can't have one without....

Some of the major terms and concepts associated with the Net are explained in this section. Arguably, you should know them well before even beginning your Web explorations, but as software like Mosaic, Cello, and Mac Web becomes increasingly popular, this is a bit like asking Windows users to keep their DOS commands in mind. It's just not going to happen. What will unquestionably happen, however, is that things will go wrong while you're cruising the Web, and without a good background knowledge of the Net you may not know what happened or how to proceed.

Domain Names

Every computer on the Internet has an Internet Protocol (IP) address associated with it. IP addresses have four parts, and a typical address looks like this: 198.43.7.85 (that is, four items all separated by periods). Happily, as a World Wide Web user you really don't need to know much about IP addresses, except possibly for getting connected to the Internet in the first place.

What you need familiarity with, however, is the Internet's domain name system (DNS). If you have an Internet account, you're already familiar with DNS: your userid, which originally looked like so much gibberish, contains the domain name for the computer on which your account exists. The U.S. president's e-mail address is `president@whitehouse.gov`, which contains the *domain name* `whitehouse.gov` and the *username* president. The domain itself is gov, which tells you that it is a government organization (big surprise), while the *subdomain,* whitehouse, tells you which part of government organization this address is attached to (again, big surprise).

Not all domain names are as easy to remember as the president's. Mine, for instance, is nrandall@watarts.uwaterloo.ca, which when analyzed reveals the following: my userid is nrandall (which at least is a whole lot more obvious than some people's userids), and my account is on a machine called watarts (the Arts faculty computer system) at the subdomain uwaterloo (the University of Waterloo), in the domain ca (which stands for the country Canada).

John December, this book's coauthor, has the address decemj@rpi.edu, which is also quite simple. His userid is decemj, the subdomain is rpi (Rensselaer Polytechnic Institute), and the domain is edu, which signifies an educational organization. In your Internet travels, you'll encounter much more complex domain names as well.

Notice the discrepancy between the domain field of my address (ca) and John's (edu). Mine points to a domain location, his to a domain type. The general rule of thumb is this: If a country code is *not* specified, assume the site is in the United States. There is in fact a domain code for the United States—not surprisingly, it's us—but it's rarely used. Similarly, a Canadian or Japanese university could use the edu domain suffix, but that also is rare. The exceptions to this, generally, are found in the net and org domains, and sometimes com, where countries are often unspecified.

Table 1.1 lists the U.S. domains, while Table 1.2 shows some of the many international domains.

Table 1.1. Domain types (usually associated with U.S. addresses).

com	Commercial organizations
edu	Educational institutions
gov	Governmental organizations (except military)
mil	Military organizations
net	Network and service providers
org	Organizations other than those listed above

Table 1.2. A sampling of international domains.

ar	Argentina
au	Australia
at	Austria

ient or-
of gov-

ince, is
userid
s), and
at the
ids for

) quite
:), and
ravels,

(edu).
s this:
fact a
Simi-
ilso is
ome-

:ional

be	Belgium
br	Brazil
ca	Canada
cl	Chile
cn	China
cr	Costa Rica
cu	Cuba
cz	Czech Republic
dk	Denmark
ec	Ecuador
eg	Egypt
fi	Finland
fr	France
de	Germany
gr	Greece
hk	Hong Kong
hu	Hungary
in	India
ir	Iran
iq	Iraq
ie	Ireland
il	Israel
it	Italy
jp	Japan
kp	North Korea
kr	South Korea
kw	Kuwait
ly	Libya
my	Malaysia
mx	Mexico
nl	Netherlands
nz	New Zealand
no	Norway

Table 1.2. continued

pa	Panama
pe	Peru
pl	Poland
pt	Portugal
pr	Puerto Rico
ro	Romania
su	Russia
lc	St. Lucia
sa	Saudi Arabia
sn	Senegal
sg	Singapore
sk	Slovakia
sl	Slovenia
za	South Africa
es	Spain
lk	Sri Lanka
se	Sweden
ch	Switzerland
sy	Syria
tw	Taiwan
th	Thailand
tr	Turkey
ua	Ukraine
ae	United Arab Emirates
uk	United Kingdom
us	United States
va	Vatican
ve	Venezuela
vn	Vietnam
zr	Zaire

Domain names affect your use of the World Wide Web in several ways. First, you're very likely to encounter something like a "DNS Lookup Error"; essentially, this means that the "domain name server" on your local computer couldn't translate the name you typed into a legitimate IP address. Practically speaking, it means that the requested file isn't available because the domain name portion of the URL was unrecognizable.

Next, the domain name is, of course, part of the entire filename itself, and you'll find yourself typing domain names plus filenames whenever you request a specific URL address. URLs, or Universal Resource Locators, contain the specific instructions for your Web browser to find and retrieve the file you specify. Clicking on a highlighted hyperlink in a Web document automatically activates the retrieval process (and effectively hides the URL address of the document from you), but Web browsers also allow you to type the URL manually.

Third, by learning domain names you can get a good sense of where you're going on the Web. In Mosaic, for example, when you place the cursor over a link, the URL address for the link appears at the bottom of the screen. It tells you immediately whether the link exists on an educational site, a commercial site, a governmental site, or something else, and this often affects your browsing strategy.

UNIX Filenames

If you want to understand the Internet thoroughly, you *must* develop a knowledge of UNIX. For a variety of reasons, mostly having to do with its strong flexibility and its excellent networking and multiuser capabilities, UNIX has become, in essence, the operating system of the Internet, and in fact all other operating systems must be customized to work with UNIX when they hook into the Net. For the time being, at least, there is no mask over the Net that makes UNIX invisible, although commercial service providers such as America Online are working hard to develop such masks.

To build a World Wide Web site, a good knowledge of UNIX is essential. To simply use the Web, however, you need to know only a tiny portion of it. An increasing number of Web users do their browsing through Macintosh or Microsoft Windows machines, and for them a very limited knowledge of UNIX is necessary. Except for one detail, in fact, they can largely do without its understanding.

That one detail is filenames. It's possible to cruise the Web without ever typing in a UNIX filename, but only if you do your browsing exclusively by clicking on hypertext links. At some point, however, you're almost guaranteed to come across an e-mail message or another document that gives you a URL (Uniform Resource Locator) address, and this almost always represents, or at least contains, a UNIX-like filename. (Actually, the URL isn't a UNIX filename; it's a standard format. But URLs tend to look very much like UNIX filenames, and understanding the filename structure will help you locate specific documents.) In every major browser, you can enter that address to move directly to that page, but if you're not exactly precise in your typing, you'll find yourself unable to get there. That's because of UNIX's complex filename structure.

Unlike, say, DOS filenames, UNIX filenames have virtually no length restriction. If you want to name a file `This_is_an_incredibly_cool.file.man`, go ahead. Of course, conventions do exist (otherwise nobody would be able to find anything), but there's nothing at all like the 8.3-character filename structure of DOS.. That's the first thing to keep in mind.

Secondly, UNIX filenames are case-sensitive. The files `OJSimpson.gif`, `OJsimpson.gif`, and `ojsimpson.gif` are all entirely different files. This is something that DOS users typically have considerable difficulty getting to handle efficiently, because DOS is completely case-insensitive. You'll find lots of UNIX filenames that combine lower case, upper case, numbers, and other symbols, and you *must* type them exactly.

Finally, the URL addresses you see will show the complete directory structure for the file. UNIX directories are very much like DOS's, except that they're separated by a forward slash (`/`) rather than a backslash (`\`). Here's a typical UNIX filename, for example: `/u2/ojsimpso/projects/dev_tools/tapp/mango_leaf.tiff`. This simply means that the file `mango_leaf.tiff` can be found in the directory `tapp`, which is a subdirectory of `dev_tools`, which is in turn a subdirectory of `projects`, which is located in the `ojsimpso` subdirectory of the `u2` directory. Not difficult to read, but awfully tricky to type.

On the Web, you'll find this in URL addresses. As already mentioned, these can be typed manually into the Open URL (or similar) dialog box in Web browsers. The following, for example, is the URL to type in when you want to access the support page for my Sams book, *Teach Yourself the Internet: Around the World in 21 Days*:

```
http://watarts.uwaterloo.ca/TYI/tyi.html
```

Similarly, if you want to access the Table of Contents page for John December's well-known *Internet Tools Summary*, type in the following URL:

```
http://www.rpi.edu/Internet/Guides/decemj/itools/toc2.html
```

In both cases, you're essentially telling your Web browser to connect to the remote server (that is, the computer where the document resides) and retrieve the specified file. Because you've done this using HTTP (HyperText Transport Protocol), the Web's standard and exclusive protocol, the file will be displayed as hypertext, complete with selectable hyperlinks. Note that both files have the extension `.html`, the standard for a page coded in Hypertext Markup Language (HTML), which is explained in detail in Part V, "Weaving a Web."

ific docu-
that page,
get there.

on. If you
urse, con-
's nothing
:o keep in

ı.gif, and
; typically
ɘtely case-
ase, num-

›r the file.
a forward
ıple: /u2/
ıt the file
ɘv_tools,
›directory

. be typed
›wing, for
my Sams

›er's well-

ɔte server
. Because
dard and
electable
ge coded
, "Weav-

FTP—File Transfer Protocol

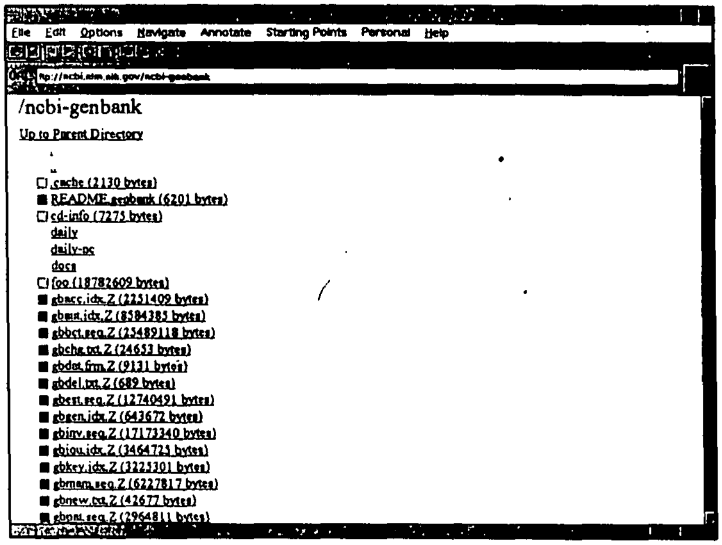
FTP is both a protocol and a program. As a protocol, it's been around almost as long as UNIX itself, and its function is to ensure a common standard for moving files from one computer to another across a network. As a program, FTP accomplishes these transfers. It enables you to enter file directories on remote machines and retrieve files from those machines or place files in those directories. A full FTP implementation offers a suite of file utilities such as creating directories and renaming and deleting files. Although you can access FTP sites by having an account with a password, the most widely used FTP type for casual users is called "anonymous FTP"; FTP software can be set to allow access to users who offer the word "anonymous" as a login name and their e-mail address as a password.

The World Wide Web makes extensive use of FTP. First, some sites do not have HTTP software in place, so to make their information accessible across the Web they place their HTML documents on FTP servers instead. Second; Web clients such as Lynx, Cello, and Mosaic make FTP connections and perform FTP downloads of files—but not uploads. In the case of sites containing graphics, sound, or video files, many are currently available via FTP only, either through anonymous FTP directly or through Gopher FTP access. FTP access is so common on the Web that you're unlikely to spend more than a few minutes cruising before you encounter an FTP transaction.

Figure 2.1. shows an FTP site as displayed through Mosaic.

FIGURE 2.1.

FTP site in Mosaic, showing extended parsing (includes file sizes).



ftp://ncbi.nlm.nih.gov/ncbi-genbank

/ncbi-genbank

Up to Parent Directory

- □ .cache (2130 bytes)
- ■ README.genbank (6201 bytes)
- □ cd-info (7275 bytes)
  - daily
  - daily-nc
  - docs
- □ foo (18782609 bytes)
- ■ gbacc.idx.Z (2251409 bytes)
- ■ gbaut.idx.Z (8584385 bytes)
- ■ gbbct.seq.Z (25489118 bytes)
- ■ gbchg.txt.Z (24653 bytes)
- ■ gbdat.frm.Z (9131 bytes)
- ■ gbdel.txt.Z (689 bytes)
- ■ gbest.seq.Z (12740491 bytes)
- ■ gbgen.idx.Z (643672 bytes)
- ■ gbinv.seq.Z (17173340 bytes)
- ■ gbjou.idx.Z (3464725 bytes)
- ■ gbkey.idx.Z (3225301 bytes)
- ■ gbmam.seq.Z (6227817 bytes)
- ■ gbnew.txt.Z (42677 bytes)
- ■ gbpri.seq.Z (2964811 bytes)

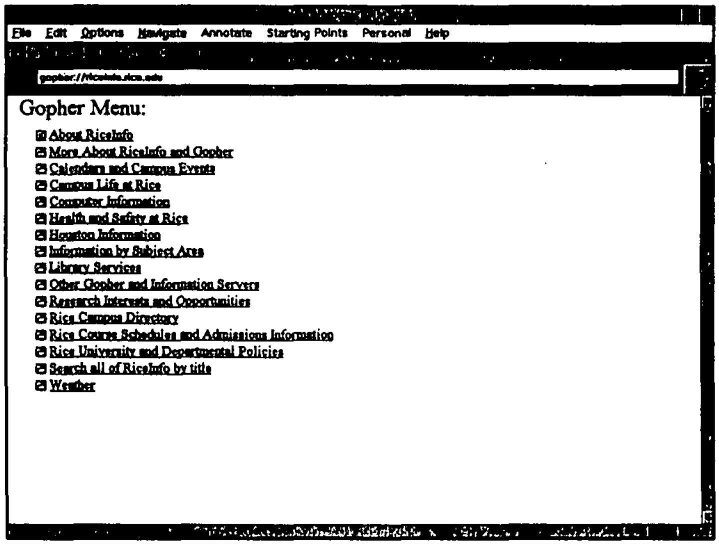
Gopher

Gopher is the best-known interface for the Internet. Developed by the University of Minnesota, Gopher is in fact very much like the World Wide Web, offering a friendly face on such difficult tools as FTP, telnet, Archie, WAIS, Veronica, and others. The main difference is that Gopher is not a hypertext environment. The most common gopher client programs present information in numbered lists instead of hyper-linked documents, with different indicators for different kinds of files (text, sound, search dialogs, etc.). Gopher software for graphical user environments such as X Window, the Macintosh, and Microsoft Windows typically offers the same lists with icons replacing the numbers. The icons offer information as to the type of file or directory you're accessing.

Gophers are directly accessible through the World Wide Web. Often this access is presented in the link itself, but you can use your browser to move to a particular gopher by specifying the `gopher://` prefix when you enter a URL address (for example, `gopher://cscns.com`). This yields the gopher directory, each item representing a selectable Web link.

Figure 2.2 shows a typical gopher directory listing as seen on the Web.

FIGURE 2.2.
Gopher directory displayed in Mosaic.



gopher://riceinfo.rice.edu

Gopher Menu:

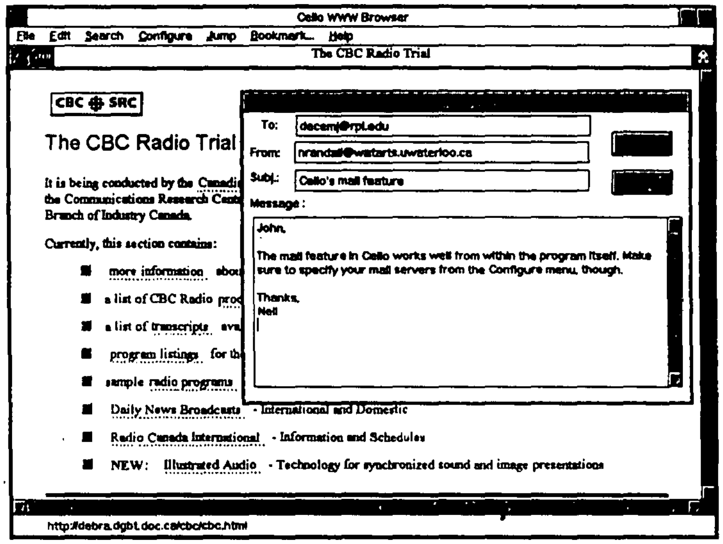
- About RiceInfo
- More About RiceInfo and Gopher
- Calendars and Campus Events
- Campus Life at Rice
- Computer Information
- Health and Safety at Rice
- Houston Information
- Information by Subject Area
- Library Services
- Other Gopher and Information Servers
- Research Interests and Opportunities
- Rice Campus Directory
- Rice Course Schedules and Admissions Information
- Rice University and Departmental Policies
- Search all of RiceInfo by title
- Weather

Electronic Mail

It's hard to be on the Internet and not know about electronic mail, but it's easy to overlook the fact that the World Wide Web is mail-enabled to a certain extent. You're not going to find a rich, full-featured e-mail package on the Web, at least not yet, but e-mail plays an important role in the Web's design. Through the use of HTML forms, you can have the readers of your pages submit mail to your address, and users of some browsers can mail directly from within that browser. (XMosaic and Cello are examples.)

Undoubtedly, the use of e-mail in the design of Web pages will increase in both usefulness and frequency. There is one main reason for this: The Web itself is primarily a public medium, while e-mail is primarily a private medium. In order to make full use of the Web even now, you often have to boot up your e-mail program and fire off messages to addresses you discover while browsing Web pages. At some point, a Web browser will probably need to include a strong e-mail feature if it is to be considered complete. Figure 2.3 shows the basic e-mail program contained within Cello.

FIGURE 2.3.
Cello's electronic mail feature.



The CBC Radio Trial

It is being conducted by the Canadi
the Communications Research Cent
Branch of Industry Canada.

Currently, this section contains:

- more information abou
- a list of CBC Radio prod
- a list of transcripts ava
- program listings for th
- sample radio programs
- Daily News Broadcasts - International and Domestic
- Radio Canada International - Information and Schedules
- NEW: Illustrated Audio - Technology for synchronized sound and image presentations

To: decemj@rpi.edu
From: nrandall@watarts.uwaterloo.ca
Subj.: Cello's mail feature
Message :

John,

The mail feature in Cello works well from within the program itself. Make sure to specify your mail servers from the Configure menu, though.

Thanks,
Neil

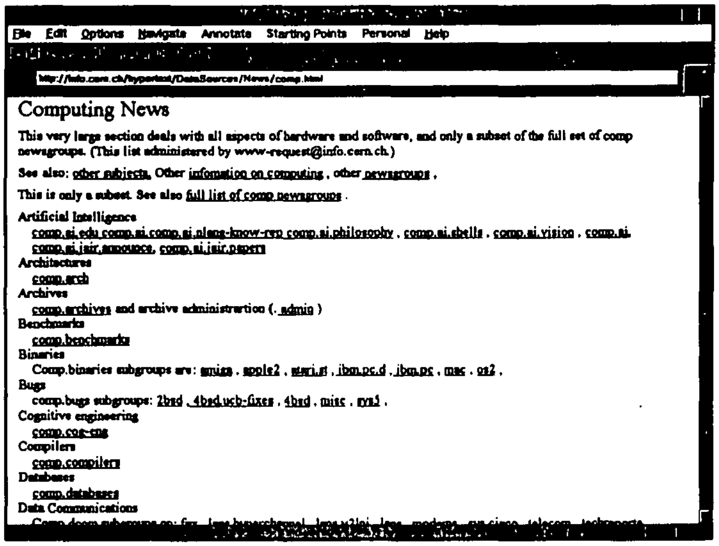
http://debra.dgbt.doc.ca/cbc/cbc.html

Usenet

Usenet has been in existence for a number of years, as the network through which users communicate in newsgroups. For some Internet users, in fact, Usenet and electronic mail are everything the Net has to offer. Usenet is the focus of any number of Internet stories in the popular press, as newsgroup users appear in stories about online romance, online harassment, the assisting of the unfortunate, and the corruption of the innocent as well. It's a very, very popular tool.

The Web has limited hooks into Usenet, but they exist and they're exploitable. Already in existence are sites that offer newsreader capability, some with graphically rendered subject "threads." It's unclear at this point how the Web will further integrate newsgroups, partly because they're so popular that it simply may not be necessary to do so. But many Web pages refer to Usenet groups as sources of additional information, so it remains important to know of their existence and their use. Keep in mind, though, that today's Web browsers offer newsreader capability only; at this stage, no browser lets you follow threads, post new messages, filter unwanted topics, or any of the other advanced capabilities of a good news program.

Figure 2.4 shows a Web link to several newsgroups.



http://info.cern.ch/hypertext/DataSources/News/comp.html

Computing News

This very large section deals with all aspects of hardware and software, and only a subset of the full set of comp newsgroups. (This list administered by www-request@info.cern.ch.)

See also: other subjects, Other information on computing , other newsgroups ,

This is only a subset. See also full list of comp newsgroups .

Artificial Intelligence
comp.ai.edu comp.ai.comp.ai.nlang-know-rep comp.ai.philosophy , comp.ai.shells , comp.ai.vision , comp.ai. comp.ai.jair.announce, comp.ai.jair.papers
Architectures
comp.arch
Archives
comp.archives and archive administrartion (. admin)
Benchmarks
comp.benchmarks
Binaries
Comp.binaries subgroups are: amiga , apple2 , atari.st , ibm.pc.d , ibm.pc , mac , os2 ,
Bugs
comp.bugs subgroups: 2bsd , 4bsd.ucb-fixes , 4bsd , misc , sys5 ,
Cognitive engineering
comp.cog-eng
Compilers
comp.compilers
Databases
comp.databases
Data Communications
Comp.dcom subgroups are: fax , lans.hyperchannel , lans.v2lni , lans , modems , sys.cisco , telecom , techreports

FIGURE 2.4.
Link through the World Wide Web to Usenet archives.

Wide Area Information Servers—WAIS

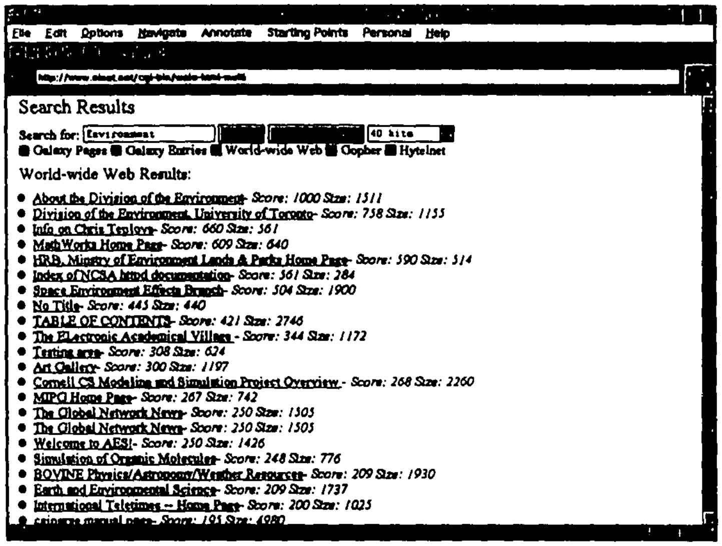
WAIS is an extremely useful tool that generates and allows you to search through a huge range of databases stored on the Internet. These databases in turn point you to locations on the Net that hold documents containing the keywords you've searched for. Among WAIS's most useful features are its relevance rating of documents—1000 means a direct hit; 100 means a marginal hit—and its ability to build, through a process called "relevance feedback," from one search to another. In other words, you can keep narrowing the search until you find exactly what you want.

The World Wide Web works through WAIS gateways to offer full keyword searching. Several pages feature WAIS searches, and in fact they've become almost the standard for finding specific titles or headings throughout the Web. Typically, WAIS searches on the Web are combined with the HTML feature called "forms," boxes that you fill in by typing text and then you execute by clicking on a button. Fully developed forms allow highly specific searches, and in some cases you can even select the areas of the Net you want searched.

The interesting part is that you often don't know that you're entering a WAIS search. As with other Internet tools, the Web has essentially co-opted the WAIS process, incorporating it into the browsers so that users can access it almost transparently.

Figure 2.5 shows a search form at the top of the screen, and the results of the WAIS search throughout the rest. Notice the relevance ratings beside the items, and note also that each item is a hyperlink to another site or document.

FIGURE 2.5.
Filling in a search form in Mosaic.



Search Results

Search for: Environment 40 hits
■ Galaxy Pages ■ Galaxy Entries ■ World-wide Web ■ Gopher ■ Hytelnet

World-wide Web Results:

- About the Division of the Environment- *Score: 1000 Size: 1511*
- Division of the Environment, University of Toronto- *Score: 758 Size: 1155*
- Info on Chris Teplovs- *Score: 660 Size: 561*
- MathWorks Home Page- *Score: 609 Size: 640*
- HRB, Ministry of Environment Lands & Parks Home Page- *Score: 590 Size: 514*
- Index of NCSA httpd documentation- *Score: 561 Size: 284*
- Space Environment Effects Branch- *Score: 504 Size: 1900*
- No Title- *Score: 445 Size: 440*
- TABLE OF CONTENTS- *Score: 421 Size: 2746*
- The ELectronic Academical Village - *Score: 344 Size: 1172*
- Testing area- *Score: 308 Size: 624*
- Art Gallery- *Score: 300 Size: 1197*
- Cornell CS Modeling and Simulation Project Overview - *Score: 268 Size: 2260*
- MIPG Home Page- *Score: 267 Size: 742*
- The Global Network News- *Score: 250 Size: 1505*
- The Global Network News- *Score: 250 Size: 1505*
- Welcome to AESI- *Score: 250 Size: 1426*
- Simulation of Organic Molecules- *Score: 248 Size: 776*
- BOVINE Physics/Astronomy/Weather Resources- *Score: 209 Size: 1930*
- Earth and Environmental Science- *Score: 209 Size: 1737*
- International Teletimes -- Home Page- *Score: 200 Size: 1025*
- [illegible] manual page- *Score: 195 Size: 4980*

Integration

These aren't the only tools accessible through the World Wide Web. Archie, Veronica, Hytelnet, and a host of others appear in Web pages from a variety of sources, and in many cases the very existence of the Web makes these tools more usable than before. None of this suggests, however, that the Web makes other manifestations of these tools obsolete. As with the newsreading capabilities of the current crop of browsers, the Web's use of the Internet's most important tools is often limited. If you need extensive features on any of these tools, you're likely better off looking elsewhere. For many users, however, the Web's versions of the tools are sufficiently capable.

Remember that the point of the World Wide Web is to offer access to the resources available on the Net. To this end, it has been designed not to replace the existing tools, but rather to integrate them into one appealing and highly usable program, taking advantage of the graphical user interface of today's machines.

There will always be those who prefer the command-line version of FTP, the richness of the feature set of a full Gopher client, and the multiple capabilities of a text- or graphics-based e-mail package. At this point, there seems little chance that any Web browser will ever fully substitute for these. Increasingly, however, the Web will integrate the Internet's tools into its structure to the degree that, for many people, the Web may well *become* the Internet. Not everyone likes that thought, but not everyone likes GUIs, voice-mail, or audio CDs, either. The point is that it's going to happen, and for anyone wishing to stay on top of the Internet, the Web will be an essential and unavoidable concept.

Fortunately, that's far from a bad thing. The Web has received its share of praise and criticism alike, but nobody denies the importance of providing solid, usable access to Internet resources. Personal computers themselves took off when their user interfaces stopped trying to emulate something out of code-level hell, and the Internet as an information provider began its rise when Archie made things findable and Gopher let you get to them. When high-speed access becomes commonly available, there's every reason to suspect that the Web will continue its trend toward becoming the most commonly accessed Internet tool—or tool*kit* — of them all.

Form PTO-1449 INFORMATION DISCLOSURE STATEMENT	Docket No.: N0003/7000	Serial No. 08/533,115
	Applicant: Glenn W. Hutton	
	Filed: September 25, 1995	Group:

U.S. Patent Documents

Ex.		Doc. No.	Date	Name	Class	Subcl.	Filed
		5309437	5/3/94	Perlman et al.	—	—	
		5442633	8/15/95	Perkins et al.	—	—	
		5166931	11/24/92	Riddle	—	—	
		5457683	10/10/95	Robins	—	—	
		5469500	11/21/95	Satter et al.	—	—	
		5452296	9/19/95	Shimizu	—	—	
		5463625	10/31/95	Yasrebi	—	—	

Foreign Patent Documents

Ex.		Doc. No.	Date	Name	Class	Subcl.	

OTHER DOCUMENTS (including, Author, Title, Date, Pages, Etc.)

Examiner:	Date considered 5/22/97

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. An * indicates references that do not require a copy to be provided under 37 C.F.R. §1.98(d) because a copy was previously cited or submitted in a prior application, which is relied upon under 35 U.S.C. §120.

#6

Form PTO-1449 INFORMATION DISCLOSURE STATEMENT	Docket No.: N0003/7000	Serial No. 08/533,115
	Applicant: Glenn W. Hutton	
	Filed: September 25, 1995	Group:

U.S. Patent Documents

Ex.		Doc. No.	Date	Name	Class	Subcl.	Filed
		5546582	8/13/96	Brockmeyer et al.	—	—	
		5430727	7/4/95	Callon	—	—	
		5524110	6/4/96	Danneels et al.	—	—	
		5095480	3/10/92	Fenner	—	—	
		5430709	7/4/95	Galloway	—	—	
		5517494	5/14/96	Green	—	—	
		5479411	12/26/95	Klein	—	—	
		5544303	8/6/96	Mraoteaux et al.	—	—	
		5526489	6/11/96	Nilakantan et al.	—	—	
		5533110	7/2/96	Pinard et al.	—	—	

Foreign Patent Documents

Ex.		Doc. No.	Date	Name	Class	Subcl.	

OTHER DOCUMENTS (including, Author, Title, Date, Pages, Etc.)

Examiner: Richard Gregson

Date considered 5/22/97

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. An * indicates references that do not require a copy to be provided under 37 C.F.R. §1.98(d) because a copy was previously cited or submitted in a prior application, which is relied upon under 35 U.S.C. §120.

#13

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :
H04J 3/14, 3/24, H04L 12/56

A1

(11) International Publication Number: **WO 92/19054**

(43) International Publication Date: 29 October 1992 (29.10.92)

(21) International Application Number: PCT/US92/02995

(22) International Filing Date: 10 April 1992 (10.04.92)

(30) Priority data:
684,695 12 April 1991 (12.04.91) US

(71) Applicant: CONCORD COMMUNICATIONS, INC. [US/US]; 753 Forest Street, Marlboro, MA 01752 (US).

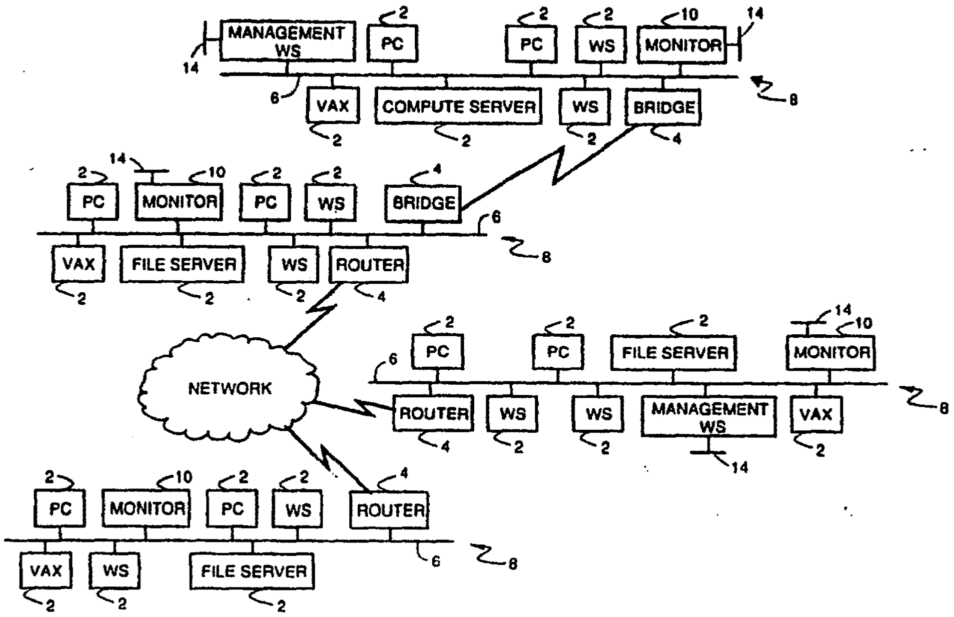
(72) Inventors: FERDINAND, Engel ; 21 Joseph Road, Northborough, MA 01532 (US). JONES, Kendall, S. ; 90 Boulder Road, Newton Center, MA 02159 (US). ROBERTSON, Kary ; 398 North Road, Bedford, MA 01739 (US). THOMPSON, David, M. ; 5127 243rd Road, Redmond, WA 98053 (US). WHITE, Gerard ; 133 Massapoag Road, Tyngsborough, MA 01879 (US).

(74) Agent: PRAHL, Eric, L.; Fish & Richardson, 225 Franklin Street, Boston, MA 02110-2804 (US).

(81) Designated States: AT (European patent), BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), MC (European patent), NL (European patent), SE (European patent).

Published
With international search report.

(54) Title: NETWORK MONITORING



(57) Abstract

Monitoring is done of communications which occur in a network of nodes (2), each communication being effected by a transmission of one or more packets among two or more communicating nodes (2), each communication complying with a predefined communication protocol selected from among protocols available in the network. The contents of packets are detected passively and in real time, communication information (130, 152, 178) associated with multiple protocols is derived from the packet contents.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CS	Czechoslovakia
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GN	Guinea
GR	Greece
HU	Hungary
IT	Italy
JP	Japan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
MC	Monaco
MC	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NL	Netherlands
NO	Norway
PL	Poland
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SN	Senegal
SU	Soviet Union
TD	Chad
TG	Togo
US	United States of America

NETWORK MONITORING

Background of the Invention

The invention relates to monitoring and managing communication networks for computers.

Todays computer networks are large complex systems with many components from a large variety of vendors. These networks often span large geographic areas ranging from a campus-like setting to world wide networks. While the network itself can be used by many different types of organizations, the purpose of these networks is to move information between computers. Typical applications are electronic mail, transaction processing, remote database, query, and simple file transfer. Usually, the organization that has installed and is running the network needs the network to be running properly in order to operate its business. Since these networks are complex systems, there are various controls provided by the different equipment to control and manage the network. Network management is the task of planning, engineering, securing and operating a network.

To manage the network properly, the Network Manager has some obvious needs. First, the Network Manager must trouble shoot problems. As the errors develop in a running network, the Network Manager must have some tools that notify him of the errors and allow him to diagnose and repair these errors. Second, the Network Manager needs to configure the network in such a manner that the network loading characteristics provide the best service possible for the network users. To do this the Network Manager must have tools that allow him visibility into access patterns, bottlenecks and general loading. With such data, the Network Manager can reconfigure the network components for better service.

There are many different components that need to be managed in the network. These elements can be, but are not limited to: routers, bridges, PC's, workstations, minicomputers, supercomputers, printers, file servers, switches and pbx's. Each component provides a protocol for reading and writing the management variables in the machine. These variables are usually defined by the component vendor and are usually referred to as a Management Information Base (MIB). There are some standard MIB's, such as the IETF (Internet Engineering Task Force) MIB I and MIB II standard definitions. Through the reading and writing of MIB variables, software in other computers can manage or control the component. The software in the component that provides remote access to the MIB variables is usually called an agent. Thus, an individual charged with the responsibility of managing a large network often will use various tools to manipulate the MIB's of various agents on the network.

Unfortunately, the standards for accessing MIBs are not yet uniformly provided nor are the MIB definitions complete enough to manage an entire network. The Network Manager must therefore use several different types of computers to access the agents in the network. This poses a problem, since the errors occurring on the network will tend to show up in different computers and the Network Manager must therefore monitor several different screens to determine if the network is running properly. Even when the Network Manager is able to accomplish this task, the tools available are not sufficient for the Network Manager to function properly.

Furthermore, there are many errors and loadings on the network that are not reported by agents. Flow control problems, retransmissions, on-off segment loading, network capacities and utilizations are some of the types of data that are not provided by the agents.

Simple needs like charging each user for actual network usage are impossible.

Summary of the Invention

In general, in one aspect, the invention features monitoring communications which occur in a network of nodes, each communication being effected by a transmission of one or more packets among two or more communicating nodes, each communication complying with a predefined communication protocol selected from among protocols available in the network. The contents of packets are detected passively and in real time, communication information associated with multiple protocols is derived from the packet contents.

Preferred embodiments of the invention include the following features. The communication information derived from the packet contents is associated with multiple layers of at least one of the protocols.

In general, in another aspect, the invention features monitoring communication dialogs which occur in a network of nodes, each dialog being effected by a transmission of one or more packets among two or more communicating nodes, each dialog complying with a predefined communication protocol selected from among protocols available in the network. Information about the states of dialogs occurring in the network and which comply with different selected protocols available in the network is derived from the packet contents.

Preferred embodiments of the invention include the following features. A current state is maintained for each dialog, and the current state is updated in response to the detected contents of transmitted packets. For each dialog, a history of events is maintained based on information derived from the contents of packets, and the history of events is analyzed to derive information about the dialog. The analysis of the history includes counting events and gathering statistics about events. The history is monitored for dialogs which are inactive, and dialogs which have been inactive for a predetermined period of time are purged. For example, the current state is updated to data state in response to observing the transmission of at least two data related packets from each node. Sequence numbers of data related packets stored in the history of events are analyzed and retransmissions are detected based on the sequence numbers. The the current state is updated based on each new packet associated with the dialog; if an updated current state cannot be determined, information about prior packets associated with the dialog is consulted as an aid in updating the state. The history of events may be searched to identify the initiator of a dialog.

The full set of packets associated with a dialog up to a point in time completely define a true state of the dialog at that point in time, and the step of updating the current state in response to the detected contents of transmitted packets includes generating a current state (e.g., "unknown") which may not conform to the true state. The current state may be updated to the true state based on information about prior packets transmitted in the dialog.

Each communication may involve multiple dialogs corresponding to a specific protocol. Each protocol layer of the communication may be parsed and analyzed to isolate each dialog and statistics may be kept for each dialog. The protocols may include a connectionless-type protocol in which the state of a dialog is implicit in transmitted packets, and the step of deriving information about the states of dialogs includes inferring the states of the dialogs from the packets. Keeping statistics for protocol layers may be temporarily suspended when parsing and statistics gathering is not rapid enough to match the rate of packets to be parsed.

In general, in another aspect, the invention features monitoring the operation of the network with respect to specific items of performance during normal operation, generating a model of the network based on the monitoring, and setting acceptable threshold levels for the specific items of performance based on the model. In preferred embodiments, the operation of the network is monitored with respect to the specific items of performance during periods which may include abnormal operation.

In general, in another aspect, the invention features the combination of a monitor connected to the network medium for passively, and in real time, monitoring transmitted packets and storing information about dialogs associated with the packets, and a workstation for receiving the information about dialogs from the monitor and providing an interface to a user. In preferred embodiments, the workstation includes means for enabling a user to observe events of active dialogs.

In general, in another aspect, the invention features apparatus for monitoring packet communications in a network of nodes in which communications may be in accordance with multiple protocols. The apparatus includes a monitor connected to a communication medium of the network for passively, and in real time, monitoring transmitted packets of different protocols and storing information about communications associated with the packets, the communications being in accordance with different protocols, and a workstation for receiving the information about the communciations from the monitor and providing an interface to a user. The monitor and the workstation include means for relaying the information about multiple protocols with respect to communication in the different protocols from the monitor to the workstation in accordance with a single common network management protocol.

In general, in another aspect, the invention features diagnosing communication problems between two nodes in a network of nodes interconnected by links. The operation of the network is monitored with respect to specific items of performance during normal operation. A model of normal operation of the network is generated based on the monitoring. Acceptable threshold levels are set for the specific items of performance based on the model. The operation of the network is monitored with respect to the specific items of performance during periods which may include abnormal operation. When abnormal operation of the network with respect to communication between the two nodes is detected, the problem is diagnosed by separately analyzing the performance of each of the nodes and each of the links connecting the two nodes to isolate the abnormal operation.

In general, in another aspect, the invention features a method of timing the duration of a transaction of interest occurring in the course of communication between nodes of a network, the beginning of the transaction being defined by the sending of a first packet of a particular kind from one node to the other, and the end of the transaction being defined by the sending of another packet of a particular kind between the nodes. In the method, packets transmitted in the network are monitored passively and in real time. The beginning time of the transaction is determined based on the appearance of the first packet. A determination is made of when the other packet has been transmitted. The timing of the duration of the transaction is ended upon the appearance of the other packet.

In general, in another aspect, the invention features, tracking node address to node name mappings in a network of nodes of the kind in which each node has a possibly nonunique node name and a unique node address within the network and in which node addresses can be assigned and reassigned to node names dynamically using a name binding protocol message incorporated within a packet. In the method, packets transmitted in the network are monitored, and a table linking node names to node addresses is updated based on information contained in the name binding protocol messages in the packets.

One advantage of the invention is that it enables a network manager to passively monitor multi-protocol networks at multiple layers of the communications. In addition, it organizes and presents network performance statistics in terms of dialogs which are occurring at any desired level of the communication. This technique of organizing and displaying network performance statistics provides an effective and useful view of network performance and facilitates a quick diagnosis of network problems.

Other advantages and features will become apparent from the following description of the preferred embodiment and from the claims.

Description of the Preferred Embodiments

Fig. 1 is a block diagram of a network;

Fig. 2 shows the layered structure of a network communication and a protocol tree within that layered environment;

Fig. 3 illustrates the structure of an ethernet/IP/TCP packet;

Fig. 4 illustrates the different layers of a communication between two nodes;

Fig. 5 shows the software modules within the Monitor;

Fig. 6 shows the structure of the Monitor software in terms of tasks and intertask communication mechanisms;

Figs. 7a-c show the STATS data structures which store performance statistics relating to the the data link layer;

Fig. 8 is a event/state table describing the operation of the state machine for a TCP connection;

Fig. 9a is a history data structure that is identified by a pointer found in the appropriate dialog statistics data within STATS;

Fig. 9b is a record from the history table;

Fig. 10 is a flow diagram of the Look_for_Data_State routine;

Fig. 11 is a flow diagram of the Look_for_Initiator routine that is called by the Look_for_Data_State routine;

Fig. 12 is a flow diagram of the Look_for_Retransmission routine which is called by the Look_at_History routine;

Fig. 13 is a diagram of the major steps in processing a frame through the Real Time Parser (RTP);

Fig. 14 is a diagram of the major steps in the processing a statistics threshold event;

Fig. 15 is a diagram of the major steps in the processing of a database update;

Fig. 16 is a diagram of the major steps in the processing of a monitor control request;

Fig. 17 is a logical map of the network as displayed by the Management Workstation;

Fig. 18 is a basic summary tool display screen;

Fig. 19 is a protocol selection menu that may be invoked through the summary tool display screen;

Figs. 20a-g are examples of the statistical variables which are displayed for different protocols;

Fig. 21 is an example of information that is displayed in the dialogs panel of the summary tool display screen;

Fig. 22 is a basic data screen presenting a rate values panel, a count values panel and a protocols seen panel;

Fig. 23 is a traffic matrix screen;

Fig. 24 is a flow diagram of the algoıithm for adaptively establishing network thresholds based upon actual network performance;

Fig. 25 is a simple multi-segment network;

Fig. 26 is a flow diagram of the operation of the diagnostic analyzer algorithm;

Fig. 27 is a flow diagram of the source node analyzer algorithm;

Fig. 28 is a flow diagram of the sink node analyzer algorithm;

Fig. 29 is a flow diagram of the link analysis logic;

Fig. 30 is a flow diagram of the DLL problem checking routine;

Fig. 31 is a flow diagram of the IP problem checking routine;

Fig. 32 is a flow diagram of the IP link component problem checking routine;

Fig. 33 is a flow diagram of the DLL link component problem checking routine;

Fig. 34 shows the structure of the event timing database;

Fig. 35 is a flow diagram of the operation of the event timing module (ETM) in the Network Monitor;

Fig. 36 is a network which includes an Appletalk® segment;

Fig. 37 is a Name Table that is maintained by the Address Tracking Module (ATM);

Fig. 38 is a flow diagram of the operation of the ATM; and

Fig. 39 is a flow diagram of the operation of the ATM.

Also attached hereto before the claims are the following appendices:

Appendix I identifies the SNMP MIB subset that is supported by the Monitor and the Management Workstation (2 pages);

Appendix II defines the extension to the standard MIB that are supported by the Monitor and the Management Workstation (25 pages);

Appendix III is a summary of the protocol variables for which the Monitor gathers statistics and a brief description of the variables, where appropriate (17 pages);

Appendix IV is a list of the Summary Tool Values Display Fields with brief descriptions (2 pages); and

Appendix V is a description of the actual screens for the Values Tool (34 pages).

Structure and Operation

The Network:

A typical network, such as the one shown in Fig. 1, includes at least three major components, namely, network nodes 2, network elements 4 and communication lines 6. Network nodes 2 are the individual computers on the network. They are the very reason the network exists. They include but are not limited to workstations (WS), personal computers (PC), file servers (FS), compute servers (CS) and host computers (e.g., a VAX), to name but a few. The term server is often used as though it was different from a node, but it is, in fact, just a node providing special services.

In general, network elements 4 are anything that participate in the service of providing data movement in a network, i.e., providing the basic communications. They include, but are not limited to, LAN's, routers, bridges, gateways, multiplexors, switches and connectors. Bridges serve as connections between different network segments. They keep track of the nodes which are connected to each of the segments to which they are connected. When they see a packet on one segment that is addressed to a node on another of their segments, they grab the packet from the one segment and transfer it to the proper segment. Gateways generally provide connections between different network segments that are operating under different protocols and serve to convert communications from one protocol to the other. Nodes send packets to routers so that they may be directed over the appropriate segments to the intended destination node.

Finally, network or communication lines 6 are the components of the network which connect nodes 2 and elements 4 together so that communicatons between nodes 2 may take place. They can be private lines, satellite lines or Public Carrier lines. They are expensive resources and are usually managed as separate entities. Often networks are organized into segments 8 that are connected by network elements 4. A segment 8 is a section of a LAN connected at a physical level (this may include repeaters). Within a segment, no protocols at layers above the physical layer are needed to enable signals from two stations on the same segment to reach each other (i.e., there are no routers, bridges, gateways...).

The Network Monitor and the Management Workstation:

In the described embodiment, there are two basic elements to the monitoring system which is to be described, namely, a Network Monitor 10 and a Management

Workstation 12. Both elements interact with each other over the local area network (LAN).

Network Monitor 10 (referred to hereinafter simply as Monitor 10) is the data collection module which is attached to the LAN. It is a high performance real time front end processor which collects packets on the network and performs some degree of analysis to search for actual or potential problems and to maintain statistical information for use in later analysis. In general, it performs the following functions. It operates in a promiscuous mode to capture and analyze all packets on the segment and it extracts all items of interest from the frames. It generates alarms to notify the Management Workstation of the occurence of significant events. It receives commands from the Management Workstation, processes them appropriately and returns responses.

Management Workstation 12 is the operator interface. It collects and presents troubleshooting and performance information to the user. It is based on the SunNet Manager (SNM) product and provides a graphical network-map-based interface and sophisticated data presentation and analysis tools. It receives information from Monitor 10, stores it and displays the information in various ways. It also instructs Monitor 10 to perform certain actions. Monitor 10, in turn, sends responses and alarms to Management Workstation 12 over either the primary LAN or a backup serial link 14 using SNMP with the MIB extensions defined later.

These devices can be connected to each other over various types of networks and are not limited to connections over a local area network. As indicated in Fig. 1, there can be multiple Workstations 12 as well as multiple Monitors 10.

Before describing these components in greater detail, background information will first be reviewed regarding communication protocols which specify how communications are conducted over the network and regarding the structure of the packets.

The Protocol Tree:

As shown in Fig. 2, communication over the network is organized as a series of layers or levels, each one built upon the next lower one, and each one specified by one or more protocols (represented by the boxes). Each layer is responsible for handling a different phase of the communication between nodes on the network. The protocols for each layer are defined so that the services offered by any layer are relatively independent of the services offered by the neighbors above and below. Although the identities and number of layers may differ depending on the network (i.e., the protocol set defining communication over the network), in general, most of them share a similar structure and have features in common.

For purposes of the present description, the Open Systems Interconnection (OSI) model will be presented as representative of structured protocol architectures. The OSI model, developed by the International Organization for Standardization, includes seven layers. As indicated in Fig. 2, there is a physical layer, a data link layer (DLL), a network layer, a transport layer, a session layer, a presentation layer and an application layer, in that order. As background for what is to follow, the function of each of these layers will be briefly described.

The physical layer provides the physical medium for the data transmission. It specifies the electrical and mechanical interfaces of the network and deals with bit level detail. The data link layer is responsible for ensuring an error-free physical link between the communicating nodes. It is responsible for creating and recognizing frame boundaries (i.e., the boundaries of the packets of data that are sent over the network.) The network layer determines how packets are routed within the network. The transport layer accepts data from the layer above it (i.e., the session layer), breaks the packets up into smaller units, if required, and passes these to the network layer for transmission over the network. It may insure that the smaller pieces all arrive properly at the other end. The session layer is the user's interface into the network. The user must interface with the session layer in order to negotiate a connection with a process in another machine. The presentation layer provides code conversion and data reformatting for the user's application. Finally, the application layer selects the overall network service for the user's application.

Fig. 2 also shows the protocol tree which is implemented by the described embodiment. A protocol tree shows the protocols that apply to each layer and it identifies by the tree structure which protocols at each layer can run "on top of" the protocols of the next lower layer. Though standard abbreviations are used to identify the protocols, for the convenience of the reader, the meaning of the abbreviations are as follows:

ARP	Address Resolution Protocol
ETHERNET	Ethernet Data Link Control
FTP	File Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
LLC	802.2 Logical Link Control
MAC	802.3 CSMA/CD Media Access Control
NFS	Network File System
NSP	Name Server Protocol
RARP	Reverse Address Resolution Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol

TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol

Two terms are commonly used to describe the protocol tree, namely, a protocol stack and a protocol family (or suite). A protocol stack generally refers to the underlying protocols that are used when sending a message over a network. For example, FTP/TCP/IP/LLC is a protocol stack. A protocol family is a loose association of protocols which tend to be used on the same network (or derive from a common source). Thus, for example, the TCP/IP family includes IP, TCP, UDP, ARP, TELNET and FTP. The Decnet family includes the protocols from Digital Equipment Corporation. And the SNA family includes the protocols from IBM.

<u>The Packet</u>:

The relevant protocol stack defines the structure of each packet that is sent over the network. Fig. 3, which shows an TCP/IP packet, illustrates the typical structure of a packet. In general, each level of the protocol stack takes the data from the next higher level and adds header information to form a protocol data unit (PDU) which it passes to the next lower level. That is, as the data from the application is passed down through the protocol layers in preparation for transmission over the network, each layer adds its own information to the data passed down from above until the complete packet is assembled. Thus, the structure of a packet ressembles that of an onion, with each PDU of a given layer wrapped within the PDU of the adjacent lower level.

At the ethernet level, the PDU includes a destination address (DEST MAC ADDR), a source address (SRC MAC ADDR), a type (TYPE) identifying the protocol which is running on top of this layer, and a DATA field for the PDU from the IP layer.

Like the ethernet packet, the PDU for the IP layer includes an IP header plus a DATA field. The IP header includes a type field (TYPE) for indicating the type of service, a length field (LGTH) for specifying the total length of the PDU, an identification field (ID), a protocol field (PROT) for identifying the protocol which is running on top of the IP layer (in this case, TCP), a source address field (SRC ADDR) for specifying the IP address of the sender, a destination address field (DEST ADDR) for specifying the IP address of the destination node, and a DATA field.

The PDU built by the TCP protocol also consists of a header and the data passed down from the next higher layer. In this case the header includes a source port field (SRC PORT) for specifying the port number of the sender, a destination port field (DEST PORT) for specifying the port number of the destination, a sequence number field (SEQ NO.) for specifying the sequence number of the data that is being sent in this packet, and an acknowledgment number field (ACK NO.) for specifying the number of the acknowledgment being returned. It also includes bits which identify the packet type, namely, an acknowledgment bit (ACK), a reset connection bit (RST), a synchronize bit (SYN), and a no more data from sender bit (FIN). There is also a window size field (WINDOW) for specifying the size of the window being used.

The Concept of a Dialog:

The concept of a dialog is used throughout the following description. As will become apparent, it is a concept which provides a useful way of conceptualizing, organizing and displaying information about the performance of a network - for any protocol and for any layer of the multi-level protocol stack.

As noted above, the basic unit of information in communication is a packet. A packet conveys meaning between the sender and the receiver and is part of a larger framework of packet exchanges. The larger exchange is called a dialog within the context of this document. That is, a dialog is a communication between a sender and a receiver, which is composed of one or more packets being transmitted between the two. There can be multiple senders and receivers which can change roles. In fact, most dialogs involve exchanges in both directions.

Stated another way, a dialog is the exchange of messages and the associated meaning and state that is inherent in any particular exchange at any layer. It refers to the exchange between the peer entities (hardware or software) in any communication. In those situations where there is a layering of protocols, any particular message exchange could be viewed as belonging to multiple dialogs. For example, in Fig. 4 Nodes A and B are exchanging packets and are engaged in multiple dialogs. Layer 1 in Node A has a dialog with Layer 1 in Node B. For this example, one could state that this is the data link layer and the nature of the dialog deals with the message length, number of messages, errors and perhaps the guarantee of the delivery. Simultaneously, Layer n of Node A is having a dialog with Layer n of node B. For the sake of the example, one could state that this is an application layer dialog which deals with virtual terminal connections and response rates. One can also assume that all of the other layers (2 through n-1) are also having simultaneous dialogs.

In some protocols there are explicit primitives that deal with the dialog and they are generally referred to as connections or virtual circuits. However, dialogs exist even in stateless and connectionless protocols. Two more examples will be described to help clarify the concept further, one dealing with a connection oriented protocol and the other dealing with a connectionless protocol.

In a typical connection oriented protocol, Node A sends a connection request (CR) message to Node B. The CR is an explicit request to form a connection. This is the start of a particular dialog, which is no different from the start of the connection. Nodes A and B could have other dialogs active simultaneously with this particular dialog. Each dialog is seen as unique. A connection is a particular type of dialog.

In a typical connectionless protocol, Node A sends Node B a message that is a datagram which has no connection paradigm, in fact, neither do the protocol(s) at higher layers. The application protocol designates this as a request to initiate some action. For example, a file server protocol such as Sun Microsystems' Network File System (NFS) could make a mount request. A dialog comes into existence once the communication between Nodes A and B has begun. It is possible to determine that communication has occurred and to determine the actions being requested. If in fact there exists more than one communication thread between Nodes A and B, then these would represent separate, different dialogs.

Inside the Network Monitor:

Monitor 10 includes a MIPS R3000 general purpose microprocessor (from MIPS Computer Systems, Inc.) running at 25 MHz. It is capable of providing 20 mips processing power. Monitor 10 also includes a 64Kbyte instruction cache and a 64Kbyte data cache, implemented by SRAM.

The major software modules of Monitor 10 are implemented as a mixture of tasks and subroutine libraries as shown in Fig. 5. It is organized this way so as to minimise the context switching overhead incurred during critical processing sequences. There is NO PREEMPTION of any module in the monitor subsystem. Each module is cognizant of the fact that it should return control to the kernel in order to let other tasks run. Since the monitor subsystem is a closed environment, the software is aware of real time constraints.

Among the major modules which make up Monitor 10 is a real time kernel 20, a boot/load module 22, a driver 24, a test module 26, an SNMP Agent 28, a Timer module 30, a real time parser (RTP) 32, a Message Transport Module (MTM) 34, a statistics database (STATS) 36, an Event Manager (EM) 38, an Event Timing Module (ETM) 40 and a control module 42. Each of these will now be described in greater detail.

Real Time Kernel 20 takes care of the general housekeeping activities in Monitor 10. It is responsible for scheduling, handling intertask communications via queues, managing a potentially large number of timers, manipulating linked lists, and handling simple memory management.

Boot/Load Module 22, which is FProm based, enables Monitor 10 to start itself when the power is turned on in the box. It initializes functions such as diagnostics, and environmental initialization and it initiates down loading of the Network Monitor Software including program and configuration files from the Management Workstation. Boot/load module 22 is also responsible for reloading program and/or configuration data following internal error detection or on command from the Management Workstation. To accomplish down loading, boot/load module 22 uses the Trivial File Transfer Protocol (TFTP). The protocol stack used for loading is TFTP/UDP/IP/ethernet over the LAN and TFTP/UDP/IP/SLIP over the serial line.

Device Driver 24 manages the network controller hardware so that Monitor 10 is able to read and write packets from the network and it manages the serial interface. It does so both for the purposes of monitoring traffic (promiscuous mode) and for the purposes of communicating with the Management Workstation and other devices on the network. The communication occurs through the network controller hardware of the physical network (e.g. Ethernet). The drivers for the LAN controller and serial line interface are used by the boot load module and the MTM. They provide access to the chips and isolate higher layers from the hardware specifics.

Test module 26 performs and reports results of physical layer tests (TDR, connectivity,...) under control of the Management Workstation. It provides traffic load information in response to user requests identifying the particular traffic data of interest. The load information is reported either as a percent of available bandwidth or as frame size(s) plus rate.

SNMP Agent 28 translates requests and information into the network management protocol being used to communicate with the Management Workstation, e.g., the Simple Network Management Protocol (SNMP).

Control Module 42 coordinates access to monitor control variables and performs actions necessary when these are altered. Among the monitor control variables which it handles are the following:

set reset monitor - transfer control to reset logic;

set time of day - modify monitor hardware clock and generate response to Management Workstation;

get time of day - read monitor hardware clock and generate response to Workstation;

set trap permit - send trap control ITM to EM and generate response to Workstation;

get trap permit - generate response to Workstation;

Control module 42 also updates parse control records within STATS when invoked by the RTP (to be described) or during overload conditions so that higher layers of parsing are dropped until the overload situation is resolved. When overload is over it restores full parsing.

Timer 30 is invoked periodically to perform general housekeeping functions. It pulses the watchdog timer at appropriate intervals. It also takes care of internal time stamping and kicking off routines like the EM routine which periodically recalculates certain numbers within the statistical database (i.e., STATS).

Real Time Parser (RTP) 32 sees all frames on the network and it determines which protocols are being used and interprets the frames. The RTP includes a protocol parser and a state machine. The protocol parser parses a received frame in the "classical" manner, layer-by-layer, lowest layer first. The parsing is performed such that the statistical objects in STATS (i.e., the network parameters for which performance data is kept) are maintained. Which layers are to have statistics stored for them is determined by a parse control record that is stored in STATS (to be described later). As each layer is parsed, the RTP invokes the appropriate functions in the statistics module (STATS) to update those statistical objects which must be changed.

The state machine within RTP 32 is responsible for tracking state as appropriate to protocols and connections. It is responsible for maintaining and updating the connection oriented statistical elements in

STATS. In order to track connection states and events, the RTP invokes a routine within the state machine. This routine determines the state of a connection based on past observed frames and keeps track of sequence numbers. It is the routine that determines if a connection is in data transfer state and if a retransmission has occurred. The objectives of the state machine are to keep a brief history of events, state transitions, and sequence numbers per connection; to detect data transfer state so that sequence tracking can begin; and to count inconsistencies but still maintain tracking while falling into an appropriate state (e.g. unknown).

RTP 32 also performs overload control by determining the number of frames awaiting processing and invoking control module 42 to update the parse control records so as to reduce the parsing depth when the number becomes too large.

Statistics Module (STATS) 36 is where Monitor 10 keeps information about the statistical objects it is charged with monitoring. A statistical object represents a network parameter for which performance information is gathered. This information is contained in an extended MIB (Management Information Base), which is updated by RTP 32 and EM 38.

STATS updates statistical objects in response to RTP invocation. There are at least four statistical object classes, namely, counters, timers, percentages (%), and meters. Each statistical object is implemented as appropriate to the object class to which it belongs. That is, each statistical object behaves such that when invoked by RTP 32 it updates and then generates an alarm if its value meets a preset threshold. (Meets means that for a high threshold the value is equal to or greater than the threshold and for a low threshold the value is equal to or less than the threshold. Note that a single object may have both high and low thresholds.)

STATS 36 is responsible for the maintenance and initial analysis of the database. This includes coordinating access to the database variables, ensuring appropriate interlocks are applied and generating alarms when thresholds are crossed. Only STATS 36 is aware of the internal structure of the database, the rest of the system is not.

STATS 36 is also responsible for tracking events of interest in the form of various statistical reductions. Examples are counters, rate meters, and rate of change of rate meters. It initiates events based on particular statistics reaching configured limits, i.e., thresholds. The events are passed to the EM which sends a trap (i.e., an alarm) to the Management Workstation. The statistics within STATS 36 are readable from the Management Workstation on request.

STATS performs lookup on all addressing fields. It assigns new data structures to address field values not currently present. It performs any hashing for fast access to the database. More details will be presented later in this document.

Event Manager (EM) 38 extracts statistics from STATS and formats it in ways that allow the Workstation to understand it. It also examines the various statistics to see if their behavior warrants a notification to the Management Workstation. If so, it uses the SNMP Agent software to initiate such notifications.

If the Workstation asks for data, EM 38 gets the data from STATS and sends it to the Workstation. It also performs some level of analysis for statistical, accounting and alarm filtering and decides on further action (e.g. delivery to the Management Workstation).

EM 38 is also responsible for controlling the delivery of events to the Management Workstation, e.g., it performs event filtering. The action to be taken on receipt of an event (e.g. threshold exceeded in STATS) is specified by the event action associated with the threshold. The event is used as an index to select the defined action (e.g. report to Workstation, run local routine xxxx, ignore). The action can be modified by commands from the Management Workstation (e.g., turn off an alarm) or by the control module in an overload situation. An update to the event action, however, does not affect events previously processed even if they are still waiting for transmission to the Management Workstation. Discarded events are counted as such by EM 38.

EM 38 also implements a throttle mechanism to limit the rate of delivery of alarms to the console based on configured limits. This prevents the rapid generation of multiple alarms. In essence, Monitor 10 is given a maximum frequency at which alarms may be sent to the Workstation. Although alarms in excess of the maximum frequency are discarded, a count is kept of the number of alarms that were discarded.

EM 38 invokes routines from the statistics module (STATS) to perform periodic updates such as rate calculations and threshold checks. It calculates time averages, e.g., average traffic by source stations, destination stations. EM 38 requests for access to monitor control variables are passed to the control module.

EM 38 checks whether asynchronous traps (i.e., alarms) to the Workstation are permitted before generating any.

EM 38 receives database update requests from the Management Workstation and invokes the statistics module (STATS) to process these.

Message Transport Module (MTM) 34, which is DRAM based, has two distinct but closely related functions. First, it is responsible for the conversion of Workstation commands and responses from the internal format used within Monitor 10 to the format used to communicate over the network. It isolates the rest of the system from the protocol used to communicate within Management Workstation. It translates between the internal representation of data and ASN.1 used for SNMP. It performs initial decoding of Workstation requests and directs the requests to appropriate modules for processing. It implements SNMP/UDP/IP/LLC or ETHERNET protocols for LAN and SNMP/UDP/IP/SLIP protocols for serial line. It receives network management commands from the Management Workstation and delivers these to the appropriate module for action. Alarms and responses destined for the Workstation are also directed via this module.

Second, MTM 34 is responsible for the delivery and reception of data to and from the Management Workstation using the protocol appropriate to the network. Primary and backup communication paths are provided transparently to the rest of the monitor modules (e.g. LAN and dial up link). It is capable of full duplex delivery of messages between the console and monitoring module. The messages carry event, configuration, test and statistics data.

Event Timing Module (ETM) 40 keeps track of the start time and end times of user specified transactions over the network. In essence, this module monitors the responsiveness of the network at any protocol or layer specified by the user.

Address Tracking Module 42 keeps track of the node name to node address bindings on networks which implement dynamic node addressing protocols.

Memory management for Monitor 10 is handled in accordance with following guidelines. The available memory is divided into four blocks during system initialization. One block includes receive frame buffers. They are used for receiving LAN traffic and for receiving secondary link traffic. These are organized as linked lists of fixed sized buffers. A second block includes system control message blocks. They are used for intertask messages within Monitor 10 and are organized as a linked list of free blocks and multiple linked lists of in process intertask messages. A third block includes transmit buffers. They are used for creation and transmission of workstation alarms and responses and are organized as a linked list of fixed sized buffers. A fourth block is the statistics. This is allocated as a fixed size area at system initialization and managed by the statistics module during system operation.

Task Structure of Monitor;

The structure of the Monitor in terms of tasks and intertask messages is shown in Fig. 6. The rectangular blocks represent interrupt service routines, the ovals represent tasks and the circles represent input queues.

Each task in the system has a single input queue which it uses to receive all input. All inter-process communications take place via messages placed onto the input queue of the destination task. Each task waits on a (well known) input queue and processes events or inter-task messages (i.e., ITM's) as they are received. Each task returns to the kernel within an appropriate time period defined for each task (e.g. after processing a fixed number of events).

Interrupt service routines (ISR's) run on receipt of hardware generated interrupts. They invoke task level processing by sending an ITM to the input queue of the appropriate task.

The kernel scheduler acts as the base loop of the system and calls any runnable tasks as subroutines. The determination of whether a task is runnable is made from the input queue, i.e., if this has an entry the task has work to perform. The scheduler scans the input queues for each task in a round robin fashion and invokes a task with input pending. Each task processes items from its input queue and returns to the scheduler within a defined period. The scheduler then continues the scan cycle of the input queues. This avoids any task locking out others by processing a continuously busy input queue. A task may be given an effectively higher priority by providing it with multiple entries in the scan table.

Database accesses are generally performed using access routines. This hides the internal structure of the database from other modules and also ensures that appropriate interlocks are applied to shared data.

The EM processes a single event from the input queue and then returns to the scheduler.

The MTM Xmit task processes a single event from its input queue and then returns control to the scheduler. The MTM Recv task processes events from the input queue until it is empty or a defined number (e.g. 10) events have been processed and then returns control to the scheduler.

The timer task processes a single event from the input queue and then returns control to the scheduler.

RTP continues to process frames until the input queue is empty or it has processed a defined number (e.g. 10) frames. It then returns to the scheduler.

The following sections contain a more detailed description of some of the above-identified software modules.

The Statistics Module (STATS):

The functions of the statistics module are:

* to define statistics records;
* to allocate and initialize statistics records;
* to provide routines to lookup statistics records, e.g. lookup_id_addr;
* to provide routines to manipulate the statistics within the records, e.g. stats_age, stats_incr and stats_rate;
* to provide routines to free statistics records, e.g. stats_allocate and stats_deallocate

It provides these services to the Real Time Parser (RTP) module and to the Event Manager (EM) module.

STATS defines the database and it contains subroutines for updating the statistics which it keeps.

STATS contains the type definitions for all statistics records (e.g. DLL, IP, TCP statistics). It provides an initialization routine whose major function is to allocate statistics records at startup from cacheable memory. It provides lookup routines in order to get at the statistics. Each type of statistics record has its own lookup routine (e.g. lookup_ip_address) which returns a pointer to a statistics record of the appropriate type or NULL.

As a received frame is being parsed, statistics within statistics records need to be manipulated (e.g. incremented) to record relevant information about the frame. STATS provides the routines to manipulate those statistics. For example, there is a routine to update counters. After the counter is incremented/decremented and if there is a non-zero threshold associated with the counter, the internal routine compares its value to the threshold. If the threshold has been exceeded, the Event Manager is signaled in order to send a trap to the Workstation. Besides manipulating statistics, these routines, if necessary, signal the Event Manager via an Intertask Message (ITM) to send a trap to the Management Workstation.

The following is an example of some of the statistics records that are kept in STATS.

o monitor statistics
o mac statistics for segment
o llc statisics for segment
o statistics per ethernet/lsap type for segment
o ip statistics for segment
o icmp statistics for segment
o tcp statistics for segment
o udp statistics for segment
o nfs statistics for segment
o ftp control statistics for segment
o ftp data statistics for segment
o telnet statistics for segment
o smtp statistics for segment
o arp statistics for segment

o statistics per mac address
o statistics per ethernet type/lasp per mac address
o statistics per ip address (includes icmp)
o statistics per tcp socket
o statistics per udp socket
o statistics per nfs socket
o statistics per ftp control socket
o statistics per ftp data socket
o statistics per telnet socket
o statistics per smtp socket
o arp statistics per ip address

o statistics per mac address pair
o statistics per ip pair (includes icmp)

- o statistics per tcp connection
- o statistics per udp pair
- o statistics per nfs pair
- o statistics per ftp control connection
- o statistics per ftp data connection
- o statistics per telnet connection
- o statistics per smtp connection

- o connection histories per udp and tcp socket

All statistics are organized similarly across protocol types. The details of the data structures for the DLL level are presented later.

As noted earlier, there are four statistical object classes (i.e., variables), namely, counts, rates, percentages (%), and meters. They are defined and implemented as follows.

A count is a continuously incrementing variable which rolls around to 0 on overflow. It may be reset on command from the user (or from software). A threshold may be applied to the count and will cause an alarm when the threshold count is reached. The threshold count fires each time the counter increments past the threshold value. For example, if the threshold is set to 5, alarms are generated when the count is 5, 10, 15,...

A rate is essentially a first derivative of a count variable. The rate is calculated at a period appropriate to the variable. For each rate variable, a minimum, maximum and average value is maintained. Thresholds may be set on high values of the rate. The maximums and minimums may be reset on command. The threshold event is triggered each time the rate calculated is in the threshold region.

As commonly used, the % is calculated at a period appropriate to the variable. For each % variable a minimum, maximum and average value is maintained. A threshold may be set on high values of the %. The threshold event is triggered each time the % calculated is in the threshold region.

Finally, a meter is a variable which may take any discrete value within a defined range. The current value has no correlation to past or future values. A threshold may be set on a maximum and/or minimum value for a meter.

The rate and % fields of network event variables are updated differently than counter or meter fields in that they are calculated at fixed intervals rather than on receipt of data from the network.

Structures for statistics kept on a per address or per address pair basis are allocated at initialization time. There are several sizes for these structures. Structures of the same size are linked together in a free pool. As a new structure is needed, it is obtained from a free queue, initialized, and linked into an active list. Active lists are kept on a per statistics type basis.

As an address or address pair (e.g. mac, ip, tcp...) is seen, RTP code calls an appropriate lookup routine. The lookup routine scans active statistics structures to see if a structure has already been allocated for the statistics. Hashing algorithms are used in order to provide for efficient lookup. If no structure has been allocated, the lookup routine examines the appropriate parse control records to determine whether statistics should be kept, and, if so, it allocates a structure of the appropriate size, initializes it and links it into an active list.

Either the address of a structure or a NULL is returned by these routines. If NULL is returned, the RTP does not stop parsing, but it will not be allowed to store the statistics for which the structure was requested.

The RTP updates statistics within the data base as it runs. This is done via macros defined for the RTP. The macros call on internal routines which know how to manipulate the relevant statistic. If the pointer to the statistics structure is NULL, the internal routine will not be invoked.

The EM causes rates to be calculated. The STATS module supplies routines (e.g. stats_rate) which must be called by the EM in order to perform the rate calculations. It also calls subroutines to reformat the data in the database in order to present it to the Workstation (i.e., in response to a get from the Workstation).

The calculation algorithms for the rate and % fields of network event variables are as follows.

The following rates are calculated in units per second, at the indicated (approximate) intervals:

1. 10 second intervals:
   e.g. DLL frame, byte, ethernet, 802.3, broadcast, multicast rates
2. 60 second intervals
   e.g., all DLL error, ethertype/dsap rates
   all IP rates.
   TCP packets, bytes, errors, retransmitted packets, retransmitted bytes, acks, rsts
   UDP packet, error, byte rates
   FTP file transfer, byte transfer, error rates

For these rates, the new average replaces the previous value directly. Maximum and minimum values are retained until reset by the user.

The following rates are calculated in units per hour at the indicated time intervals:

1. 15 minute interval.

e.g., TCP - connection rate
Telnet connection rate
FTP session rate

The hourly rate is calculated from a sum of the last twelve 5 minute readings, as obtained from the buckets for the pertinent parameter. Each new reading replaces the oldest of the twelve values maintained. Maximum and minimum values are retained until reset by the user.

There are a number of other internal routines in STATS. For example, all statistical data collected by the Monitor is subject to age out. Thus, if no activity is seen for an address (or address pair) in the time period defined for age out, then the data is discarded and the space reclaimed so that it may be recycled. In this manner, the Monitor is able to use the memory for active elements rather than stale data. The user can select the age out times for the different components. The EM periodically kicks off the aging mechanism to perform this recycling of resources. STATS provides the routines which the EM calls, e.g. stats_age.

There are also routines in STATS to allocate and de-allocate Statistics, e.g., stats_allocate and stats_de-allocate. The allocate routine is called when stations and dialogs are picked up by the Network Monitor. The de-allocate routine is called by the aging routines when a structure is to be recycled.

The Data Structures in STATS

The general structure of the database within STATS is illustrated by Figs. 7a-c, which shows information that is maintained for the Data Link Layer (DLL) and its organization. A set of data structures is kept for each address associated with the layer. In this case there are three relevant addresses, namely a segment address, indicating which segment the node is on, a MAC address for the node on the segment, and an address which identifies the dialog occurring over that layer. The dialog address is the combination of the MAC addresses for the two nodes which make up the dialog. Thus, the overall data structure has three identifiable components: a segment address data structure (see Fig. 7a), a MAC address data structure (see Fig. 7b) and a dialog data structure (see Fig. 7c).

The segment address structure includes a doubly linked list 102 of segment address records 104, each one for a different segment address. Each segment address record 104 contains a forward and backward link (field 106) for forward and backward pointers to neighboring records and a hash link (field 108). In other words, the segment address records are accessed by either walking down the doubly linked list or by using a hashing mechanism to generate a pointer into the doubly linked list to the first record of a smaller hash linked list. Each record also contains the address of the segment (field 110) and a set of fields for other information. Among these are a flags field 112, a type field 114, a parse_control field 116, and an EM_control field 118. Flags field 112 contains a bit which indicates whether the identified address corresponds to the address of another Network Monitor. This field only has meaning in the MAC address record and not in the segment or dialog address record. Type field 114 identifies the MIB group which applies to this address. Parse control field 116 is a bit mask which indicates what subgroups of statistics from the identified MIB group are maintained, if any. Flags field 112, type field 114 and parse control field 116 make up what is referred to as the parse control record for this MAC address. The Network Monitor uses a default value for parse control field 116 upon initialization or whenever a new node is detected.

The default value turns off all statistics gathering. The statistics gathering for any particular address may subsequently be turned on by the Workstation through a Network Monitor control command that sets the appropriate bits of the parse control field to one.

EM_control field 118 identifies the subgroups of statistics within the MIB group that have changed since the EM last serviced the database to update rates and other variables. This field is used by the EM to identify those parts of STATS which must be updated or for which recalculations must be performed when the EM next services STAT.

Each segment address record 104 also contains three fields for time related information. There is a start_time field 120 for the time that is used to perform some of the rate calculations for the underlying statistics; a first_seen field 122 for the time at which the Network Monitor first saw the communication; and a last_seen field 124 for the time at which the last communication was seen. The last_seen time is used to age out the data structure if no activity is seen on the segment after a preselected period of time elapses. The first_seen time is a statistic which may be of interest to the network manager and is thus retrievable by the Management Workstation for display.

Finally, each segment address record includes a stats_pointer field 126 for a pointer to a DLL segment statistics data structure 130 which contains all of the statistics that are maintained for the segment address. If the bits in parse_control field 116 are all set to off, indicating that no statistics are to be maintained for the address, then the pointer in stats_pointer field 126 is a null pointer.

The list of events shown in data structure 130 of Fig. 7a illustrates the type of data that is collected for this address when the parse control field bits are set to on. Some of the entries in DLL segment statistics data structure 130 are pointers to buckets for historical data. In the case where buckets are maintained, there are twelve buckets each of which represents a time period of five minutes duration and each of which generally contains two items of information, namely, a count for the corresponding five minute time period and a MAX rate for that time period. MAX rate records any spikes which have occurred during the period and which the user may not have observed because he was not viewing that particular statistic at the time.

At the end of DLL segment statistics data structure 130, there is a protocol_Q pointer 132 to a linked list 134 of protocol statistics records 136 identifying all of the protocols which have been detected running on top of the DLL layer for the segment. Each record 136 includes a link 138 to the next record in the list, the identity of the protocol (field 140), a frames count for the number of frames detected`for the identified protocol (field 142); and a frame rate (field 144).

The MAC address data structure is organized in a similar manner to that of the segment data structure (see Fig. 7b). There is a doubly linked list 146 of MAC address records 148, each of which contains the same type of information as is stored in DLL segment address records 104. A pointer 150 at the end of each MAC address record 148 points to a DLL address statistics data structure 152, which like the DLL segment address data structure 130, contains fields for all of the statitics that are gathered for that DLL MAC address. Examples of the particular statistics are shown in Fig. 7b.

At the end of DLL address statistics data structure 152, there are two pointer fields 152 and 154, one for a pointer to a record 158 in a dialog link queue 160, and the other for a pointer to a linked list 162 of protocol statistics records 164. Each dialog link queue entry 158 contains a pointer to the next entry (field 168) in the queue and it contains a dialog_addr pointer 170 which points to an entry in the DLL dialog queue which involves the MAC address. (see Fig. 7c). Protocol statistics records 164 have the same structure and contain the same categories of information as their counterparts hanging off of DLL segment statistics data structure 130.

The above-described design is repeated in the DLL dialog data structures. That is, dialog record 172 includes the same categories of information as its counterpart in the DLL segment address data structure and the MAC address data structure. The address field 174 contains the addresses of both ends of the dialog concatenated together to form a single address. The first and second addresses within the single address are arbitrarily designated nodes 1 and 2, respectively. In the stats_pointer field 176 there is a pointer to a dialog statistics data structure 178 containing the relevant statistics for the dialog. The entries in the first two fields in this data structure (i.e., fields 180 and 182) are designated protocol entries and protocols. Protocol entries is the number of different protocols which have been seen between the two MAC addresses. The protocols that have been seen are enumerated in the protocols field 182.

DLL dialog statistics data structure 178, illustrated by Fig. 7c, includes several additional fields of information which only appear in these structures for dialogs for which state information can be kept (e.g. TCP connection). The additional fields identify the transport protocol (e.g., TCP) (field 184) and the application which is running on top of that protocol (field 186). They also include the identity of the initiator of the connection (field 188), the state of the connection (field 190) and the reason that the connection was closed, when it is closed (field 192). Finally, they also include a state_pointer (field 194) which points to a history data structure that will be described in greater detail later. Suffice it to say, that the history data structure contains a short history of events and states for each end of the dialog. The state machine uses the information contained in the history data structure to loosely determine what the state of each of the end nodes is throughout the course of the connection. The qualifier "loosely" is used because the state machine does not closely shadow the state of the connection and thus is capable of recovering from loss of state due to lost packets or missed communications.

The above-described structures and organization are used for all layers and all protocols within STATS.

Real Time Parser (RTP)

The RTP runs as an application task. It is scheduled by the Real Time Kernel scheduler when received frames are detected. The RTP parses the frames and causes statistics, state tracking, and tracing operations to be performed.

The functions of the RTP are:

- obtain frames from the RTP Input Queue;
- parse the frames;
- maintain statistics using routines supplied by the STATS module;
- maintain protocol state information;

- * notify the MTM via an ITM if a frame has been received with the Network Monitor's address as the destination address; and
- * notify the EM via an ITM if a frame has been received with any Network Monitor's address as the source address.

The design of the RTP is straightforward. It is a collection of routines which perform protocol parsing. The RTP interfaces to the Real Time Kernel in order to perform RTP initialization, to be scheduled in order to parse frames, to free frames, to obtain and send an ITM to another task; and to report fatal errors. The RTP is invoked by the scheduler when there is at least one frame to parse. The appropriate parse routines are executed per frame. Each parse routine invokes the next level parse routine or decides that parsing is done. Termination of the parse occurs on an error or when the frame has been completely parsed.

Each parse routine is a separately compilable module. In general, parse routines share very little data. Each knows where to begin parsing in the frame and the length of the data remaining in the frame.

The following is a list of the parse routines that are available within RTP for parsing the different protocols at the various layers.

Data Link Layer Parse - rtp_dll_parse:

This routine handles Ethernet, IEEE 802.3, IEEE 802.2, and SNAP. See RFC 1010, Assigned Numbers for a description of SNAP (Subnetwork Access Protocol).

Address Resolution Protocol Parse - rtp_arp_parse

ARP is parsed as specified in RFC 826.

Internet Protocol Parse - rtp_ip_parse

IP Version 4 is parsed as specified in RFC 791 as amended by RFC 950, RFC 919, and RFC 922.

Internet Control Message Protocol Parse - rtp_icmp_parse

ICMP is parsed as specified in RFC 792.

Unit Data Protocol Parse - rtp_udp_parse

UDP is parsed as specified in RFC 768.

Transmission Control Protocol Parse - rtp_tcp_parse

TCP is parsed as specified in RFC 793.

Simple Mail Transfer Protocol Parse - rtp_smtp_parse

SMTP is parsed as specified in RFC 821.

File Transfer Protocol Parse - rtp_ftp_parse

FTP is parsed as specified in RFC 959.

Telnet Protocol Parse - rtp_telnet_parse

The Telnet protocol is parsed as specified in RFC 854.

Network File System Protocol Parse - rpt_nfs_parse

The NFS protocol is parsed as specified in RFC 1094.

The RTP calls routines supplied by STATS to look up data structures. By calling these lookup routines, global pointers to data structures are set up. Following are examples of the pointers to statistics data structures that are set up when parse routines call Statistics module lookup routines.

mac_segment, mac_dst_segment, mac_this_segment,
mac_src, mac_dst, mac_dialog
ip_src_segment, ip_dst_segment, ip_this_segment,
ip_src, ip_dst, ip_dialog
tcp_src_segment, tcp_dst_segment,
tcp_this_segment,
tcp_src, tcp_dst, tcp_src_socket, tcp_dst_socket,
    tcp_connection

The mac_src and mac_dst routines return pointers to the data structures within STATS for the source MAC address and the destination MAC address, respectively. The lookup_mac_dialog routine returns a pointer to the data structure within STATS for the dialog between the two nodes on the MAC layer. The other STATS routines supply similar pointers for data structures relevant to other protocols.

The RTP routines are aware of the names of the statistics that must be manipulated within the data base (e.g. frames, bytes) but are not aware of the structure of the data. When a statistic is to be manipulated, the RTP routine invokes a macro which manipulates the appropriate statistics in data structures. The macros use the global pointers which were set up during the lookup process described above.

After a frame has been parsed (whether the parse was successful or not), the RTP routine examines the destination mac and ip addresses. If either of the addresses is that of the Network Monitor, RTP obtains a low priority ITM, initializes it, and sends the ITM to the MTM task. One of the fields of the ITM contains the address of the buffer containing the frame.

The RTP must hand some received frames to the EM in order to accomplish the autotopology function (described later). After a frame has been parsed (whether the parse was successful or not), the RTP routine examines the source mac and ip addresses. If either of the addresses is that of another Network Monitor, RTP obtains a low priority ITM, initializes it and sends the ITM to the EM task. The address data structure (in particular, the flags field of the parse control record) within STATS for the MAC or the IP address indicates whether the source address is that of another Network Monitor. One of the fields of the ITM contains the address of the buffer containing the frame.

The RTP receives traffic frames from the network for analysis. RTP operation may be modified by sending control messages to the Monitor. RTP first parses these messages, then detects that the messages are destined for the Monitor and passes them to the MTM task. Parameters which affect RTP operation may be changed by such control messages.

The general operation of the RTP upon receipt of a traffic frame is as follows:

```
Get next frame from input queue
get address records for these stations
For each level of active parsing
{
get pointer to start of protocol header
call layer parse routine
determine protocol at next level
set pointer to start of next layer protocol

}end of frame parsing
if this is a monitor command add to MTM input
queue
if this frame is from another monitor, pass
to EM
check for overload -if yes tell control
```

The State Machine:

In the described embodiment, the state machine determines and keeps state for both addresses of all TCP connections. TCP is a connection oriented transport protocol, and TCP clearly defines the connection in terms of states of the connection. There are other protocols which do not explicitly define the communication in terms of state, e.g. connectionless protocols such as NFS. Nevertheless, even in the connectionless protocols there is implicitly the concept of state because there is an expected order to the events which will occur during the course of the communication. That is, at the very least, one can identify a beginning and an end of the communication, and usually some sequence of events which will occur during the course of the communication. Thus, even though the described embodiment involves a connection oriented protocol, the principles are applicable to many connectionless protocols or for that matter any protocol for which one can identify a beginning and an end to the communication under that protocol.

Whenever a TCP packet is detected, the RTP parses the information for that layer to identify the event associated with that packet. It then passes the identified event along with the dialog identifier to the state machine. For each address of the two parties to the communication, the state machine determines what the current state of the node is. The code within the state machine determines the state of a connection based upon a set of rules that are illustrated by the event/state table shown in Fig. 8.

The interpretation of the event/state table is as follows. The top row of the table identifies the six possible states of a TCP connection. These states are not the states defined in the TCP protocol specification. The left most column identifies the eight events which may occur during the course of a connection. Within the table is an array of boxes, each of which sits at the intersection of a particular event/state combination. Each box specifies the actions taken by the state machine if the identified event occurs while the connection is in the identified state. When the state machine receives a new event, it may perform three types of action. It may change the recorded state for the node. The state to which the node is changed is specified by the S="STATE" entry located at the top of the box. It may increment or decrement the appropriate counters to record the information relevant to that event's occurrence. (In the table, incrementing and decrementing are signified by the ++ and the -- symbols, respectively, located after the identity of the variable being updated.) Or the state machine may take other actions such as those specified in the table as start close timer, Look_for_Data_State, or Look_at_History (to be described shortly). The particular actions which the state machine takes are specified in each box. An empty box indicates that no action is taken for that particular event/state combination. Note, however, that the occurrence of an event is also likely to have caused the update of statistics within STATS, if not by the state machine, then by some other part of the RTP. Also note that it may be desirable to have the state machine record other events, in which case the state table would be modified to identify those other actions.

Two events appearing on the table deserve further explanation, namely, close timer expires and inactivity timer expires. The close timer, which is specified by TCP, is started at the end of a connection and it establishes a period during which any old packets for the connection which are received are thrown away (i.e., ignored). The inactivity timer is not specified by TCP but rather is part of the Network Monitor's resource management functions. Since keeping statistics for dialogs (especially old dialogs) consumes resources, it is desirable to recycle resources for a dialog if no activity has been seen for some period of time. The inactivity timer provides the mechanism for accomplishing this. It is restarted each time an event for the connection is received. If the inactivity timer expires (i.e., if no event is received before the timer period ends), the connection is assumed to have gone inactive and all of the resources associated with the dialog are recycled. This involves freeing them up for use by other dialogs.

The other states and events within the table differ from but are consistent with the definitions provided by TCP and should be self evident in view of that protocol specification.

The event/state table can be read as follows. Assume, for example, that node 1 is in DATA state and the RTP receives another packet from node 1 which it determines to be a TCP FIN packet. According to the entry in the table at the intersection of FIN/DATA (i.e., event/state), the state machine sets the state of the connection for node 1 to CLOSING, it decrements the active connections counter and it starts the close timer. When the close timer expires, assuming no other events over that connection have occurred, the state machine sets node 1's state to CLOSED and it starts the inactivity timer. If the RTP sends another SYN packet to reinitiate a new connection before the inactive timer expires, the state machine sets node 1's state to CONNECTING (see the SYN/CLOSED entry) and it increments an after close counter.

When a connection is first seen, the Network Monitor sets the state of both ends of the connection to UNKNOWN state. If some number of data and acknowledgment frames are seen from both connection ends, the states of the connection ends may be promoted to DATA state. The connection history is searched to make this determination as will be described shortly.

Referring to Figs. 9a-b, within STATS there is a history data structure 200 which the state machine uses to remember the current state of the connection, the state of each of the nodes participating in the connection and a short history of state related information. History data structure 200 is identified by a state_pointer found at the end of the associated dialog statistics data structure in STATS (see Fig. 7c). Within history data structure 200, the state machine records the current state of node 1 (field 202), the current state of node 2 (field 206) and other data relating to the corresponding node (fields 204 and 208). The other data includes, for example, the window size for the receive and transmit communications, the last detected sequence numbers for the data and acknowledgment frames, and other data transfer information.

History data structure 200 also includes a history table (field 212) for storing a short history of events which have occurred over the connection and it includes an index to the next entry within the history table for storing the information about the next received event (field 210). The history table is implemented as a circular buffer which includes sufficient memory to store, for example, 16 records. Each record, shown in Fig. 9b, stores the state of the node when the event was detected (field 218), the event which was detected (i.e., received) (field 220), the data field length (field 222), the sequence number (field 224), the acknowledgment sequence number (field 226) and the identity of the initiator of the event, i.e., either node 1 or node 2 or 0 if neither (field 228).

Though the Network Monitor operates in a promiscuous mode, it may occasionally fail to detect or it may, due to overload, lose a packet within a communication. If this occurs the state machine may not be able to accurately determine the state of the connection upon receipt of the next event. The problem is evidenced by the fact that the next event is not what was expected. When this occurs, the state machine tries to recover state by relying on state history information stored in the history table in field 212 to deduce what the state is. To deduce the current state from historical information, the state machine uses one of the two previously mentioned routines, namely, Look_for_Data_State and Look_at_History.

Referring to Fig. 10, Look_for_Data_State routine 230 searches back through the history one record at a time until it finds evidence that the current state is DATA state or until it reaches the end of the circular buffer (step 232). Routine 230 detects the existence of DATA state by determining whether node 1 and node 2 each have had at least two data events or two acknowledgment combinations with no intervening connect, disconnect or abort events (step 234). If such a sequence of events is found within the history, routine 230 enters both node 1 and node 2 into DATA state (step 236), it increments the active connections counter (step 238) and then it calls a Look_for_Initiator routine to look for the initiator of the connection (step 240). If such a pattern of events is not found within the history, routine 230 returns without changing the state for the node (step 242).

As shown in Fig. 11, Look_for_Initiator routine 240 also searches back through the history to detect a telltale event pattern which identifies the actual initiator of the connection (step 244). More specifically, routine 240 determines whether nodes 1 and 2 each sent connect-related packets. If they did, routine 240 identifies the initiator as the first node to send a connect-related packet (step 246). If the search is not successful, the identity of the connection initiator remains unknown (step 248).

The Look_at_History routine is called to check back through the history to determine whether data transmissions have been repeated. In the case of retransmissions, the routine calls a Look_for_Retransmission routine 250, the operation of which is shown in Fig. 12. Routine 250 searches back through the history (step 252) and checks whether the same initiator node has sent data twice (step 254). It detects this by comparing the current sequence number of the packet as provided by the RTP with the sequence numbers of data packets that were previously sent as reported in the history table. If a retransmission is spotted, the retransmission counter in the dialog statistics data structure of STATS is incremented (step 256). If the sequence number is not found within the history table, indicating that the received packet does not represent a retransmission, the retransmission counter is not incremented (step 258).

Other statistics such as Window probes and keep alives may also be detected by looking at the received frame, data transfer variables, and, if necessary, the history.

Even if frames are missed by the Network Monitor, because it is not directly "shadowing" the connection, the Network Monitor still keeps useful statistics about the connection. If inconsistencies are detected the Network Monitor counts them and, where appropriate, drops back to UNKNOWN state. Then, the Network Monitor waits for the connection to stabilize or deteriorate so that it can again determine the appropriate state based upon the history table.

Principal Transactions of Network Monitor Modules:

The transactions which represent the major portion of the processing load within the Monitor, include monitoring, actions on threshold alarms, processing database get/set requests from the Management Workstation, and processing monitor control requests from the Management Workstation. Each of these mechanisms will now be briefly described.

Monitoring involves the message sequence shown in Fig. 13. In that figure, as in the other figures involving message sequences, the numbers under the heading SEQ. identify the major steps in the sequence. The following steps occur:

1. ISR puts Received traffic frame ITM on RTP input queue
2. request address of pertinent data structure from STATS (get parse control record for this station)
3. pass pointer to RTP
4. update statistical objects by call to statistical update routine in STATS using pointer to pertinent data structure
5. parse completed - release buffers

The major steps which follow a statistics threshold event (i.e., an alarm event) are shown in Fig. 14. The steps are as follows:

1. statistical object update causes threshold alarm
2. STATS generates threshold event ITM to event manager (EM)
3. look up appropriate action for this event
4. perform local event processing
5. generate network alarm ITM to MTM Xmit (if required)
6. format network alarm trap for Workstation from event manager data
7. send alarm to Workstation

The major steps in processing of a database update request (i.e., a get/set request) from the Management Workstation are shown in Fig. 15. The steps are as follows:

1. LAN ISR receives frame from network and passes it to RTP for parsing
2. RTP parses frame as for any other traffic on segment.
3. RTP detects frame is for monitor and sends received Workstation message over LAN ITM to MTM Recv.

4. MTM Recv processes protocol stack.
5. MTM Recv sends database update request ITM to EM.
6. EM calls STATS to do database read or database write with appropriate IMPB
7. STATS performs database access and returns response to EM.
8. EM encodes response to Workstation and sends database update response ITM to MTM Xmit
9. MTM Xmit transmits.

The major steps in processing of a monitor control request from the Management Workstation are shown in Fig. 16. The steps are as follows:

1. Lan ISR receives frame from network and passes received frame ITM to RTP for parsing.
2. RTP parses frame as for any other traffic on segment.
3. RTP detects frame is for monitor and sends received workstation message over LAN ITM to MTM Recv.
4. MTM Recv processes protocol stack and decodes workstation command.
5. MTM Recv sends request ITM to EM.
6. EM calls Control with monitor control IMPB.
7. Control performs requested operation and generates response to EM.
8. EM sends database update response ITM to MTM Xmit.
9. MTM Xmit encodes response to Workstation and transmits.

The Monitor/Workstation Interface:

The interface between the Monitor and the Management Workstation is based on the SNMP definition (RFC 1089 SNMP; RFC 1065 SMI; RFC 1066 SNMP MIB - Note: RFC means Request for Comments). All five SNMP PDU types are supported:

get-request

get-next-request
get-response
set-request
trap

The SNMP MIB extensions are designed such that where possible a user request for data maps to a single complex MIB object. In this manner, the get-request is simple and concise to create, and the response should contain all the data necessary to build the screen. Thus, if the user requests the IP statistics for a segment this maps to an IP Segment Group.

The data in the Monitor is keyed by addresses (MAC, IP) and port numbers (telnet, FTP). The user may wish to relate his data to physical nodes entered into the network map. The mapping of addresses to physical nodes is controlled by the user (with support from the Management Workstation system where possible) and the Workstation retains this information so that when a user requests data for node 'Joe' the Workstation asks the Monitor for the data for the appropriate address(es). The node to address mapping need not be one to one.

Loading and dumping of monitors uses TFTP (Trivial File Transfer Protocol). This operates over UDP as does SNMP. The Monitor to Workstation interface follows the SNMP philosophy of operating primarily in a polled mode. The Workstation acts as the master and polls the Monitor slaves for data on a regular (configurable) basis.

The information communicated by the SNMP is represented according to that subset of ASN.1 (ISO 8824 Specification of ASN.1) defined in the Internet standard Structure of Management Information (SMI - RFC 1065). The subset of the standard Management Information Base (MIB) (RFC 1066 SNMP MIB) which is supported by the Workstation is defined in Appendix III. The added value provided by the Workstation is encoded as enterprise specific extensions to the MIB as defined in Appendix IV. The format for these extensions follows the SMI recomendations for object identifiers so that the Workstation extensions fall in the subtree 1.3.6.1.4.1.x.1. where x is an enterprise specific node identifier assigned by the IAB.

Appendix V is a summary of the network variables for which data is collected by the Monitor for the extended MIB and which can be retrieved by the Workstation. The summary includes short decriptions of the meaning and significance of the variables, where appropriate.

The Management Workstation:

The Management Workstation is a SUN Sparcstation (also referred to as a Sun) available from Sun Microsystems, Inc. It is running the Sun flavor of Unix and uses the Open Look Graphical User Interface (GUI) and the SunNet Manager as the base system. The options required are those to run SunNet Manager with some additional disk storage requirement.

The network is represented by a logical map illustrating the network components and the relationships between them, as shown in Fig. 17. A hierarchical network map is supported with navigation through the layers of the hierarchy, as provided by SNM. The Management Workstation determines the topology of the network and informs the user of the network objects and their connectivity so that he can create a network map. To assist with the map creation process, the Management Workstation attempts to determine the stations connected to each LAN segment to which a Monitor is attached. Automatic determination of segment topology by detecting stations is performed using the autotopology algorithms as described in copending U.S. Patent Application S.N. ***,*** entitled "Automatic Topology Monitor for Multi-

Segment Local Area Network" filed on January 14, 1991 (Attorney Docket No. 13283-NE.APP), incorporated herein by reference.

In normal operation, each station in the network is monitored by a single Monitor that is located on its local segment. The initial determination of the Monitor responsible for a station is based on the results of the autotopology mechanism. The user may override this initial default if required.

The user is informed of new stations appearing on any segment in the network via the alarm mechanism. As for other alarms, the user may select whether stations appearing on and disappearing from the network segment generate alarms and may modify the times used in the aging algorithms. When a new node alarm occurs, the user must add the new alarm to the map using the SNM tools. In this manner, the SNM system becomes aware of the nodes.

The sequence of events following the detection of a new node is:

1. the location of the node is determined automatically for the user.
2. the Monitor generates an alarm for the user indicating the new node and providing some or all of the following information:
   - mac address of node
   - ip address of node
   - segment that the node is believed to be
     located on
   - Monitor to be responsible for the node
3. the user must select the segment and add the node manually using the SNM editor

4. The update to the SNM database will be detected and the file reread. The Workstation database is reconstructed and the parse control records for the Monitors updated if required.
5. The Monitor responsible for the new node has its parse control record updated via SNMP set request(s).

An internal record of new nodes is required for the autotopology. When a new node is reported by a Network Monitor, the Management Workstation needs to have the previous location information in order to know which Network Monitors to involve in autotopology. For example, two nodes with the same IP address may exist in separate segments of the network. The history makes possible the correlation of the addresses and it makes possible duplicate address detection.

Before a new Monitor can communicate with the Management Workstation via SNMP it needs to be added to the SNM system files. As the SNM files are cached in the database, the file must be updated and the SNM system forced to reread it.

Thus, on the detection of a new Monitor the following events need to occur in order to add the Monitor to the Workstation:

1. The Monitor issues a trap to the Management Workstation software and requests code to be loaded from the Sun Microsystems boot/load server.
2. The code load fails as the Monitor is not known to the unix networking software at this time.
3. The Workstation confirms that the new Monitor does not exceed the configured system limits (e.g. 5 Monitors per

Workstation) and terminates the initialization sequence if limits are exceeded. An alarm is issued to the user indicating the presence of the new Monitor and whether it can be supported.

4. The user adds the Monitor to the SNMP.HOSTS file of the SNM system, to the etc/hosts file of the Unix networking system and to the SNM map.
5. When the files have been updated the user resets the Monitor using the set tool (described later).
6. The Monitor again issues a trap to the Management Workstation software and requests code to be loaded from the Sun boot/load server.
7. The code load takes place and the Monitor issues a trap requesting data from the Management Workstation.
8. The Monitor data is issued using SNMP set requests.

Note that on receiving the set request, the SNMP proxy rereads in the (updated) SNMP.HOSTS file which now includes the new Monitor. Also note that the SNMP hosts file need only contain the Monitors, not the entire list of nodes in the system.

9. On completion of the set request(s) the Monitor run command is issued by the Workstation to bring the Monitor on line.

The user is responsible for entering data into the SNM database manually. During operation, the Workstation monitors the file write date for the SNM database. When this is different from the last date read, the SNM database is reread and the Workstation database reconstructed. In this manner, user updates to the SNM database are incorporated into the Workstation database as quickly as possible without need for the user to take any action.

When the Workstation is loaded, the database is created from the data in the SNM file system (which the user has possibly updated). This data is checked for consistency and for conformance to the limits imposed by the Workstation at this time and a warning is generated to the user if any problems are seen. If the data errors are minor the system continues operation; if they are fatal the user is asked to correct them and Workstation operation terminates.

The monitoring functions of the Management Workstation are provided as an extension to the SNM system. They consist of additional display tools (i.e., summary tool, values tool, and set tool) which the user invokes to access the Monitor options and a Workstation event log in which all alarms are recorded.

As a result of the monitoring process, the Monitor makes a large number of statistics available to the operator. These are available for examination via the Workstation tools that are provided. In addition, the Monitor statistics (or a selected subset thereof) can be made visible to any SNMP manager by providing it with knowledge of the extended MIB. A description of the statistics maintained are described elswhere.

Network event statistics are maintained on a per network, per segment and per node basis. Within a node, statistics are maintained on a per address (as appropriate to the protocol layer - IP address, port number, ...) and per connection basis. Per network statistics are always derived by the Workstation from the per segment variables maintained by the Monitors. Subsets of the basic statistics are maintained on a node to node and segment to segment basis.

If the user requests displays of segment to segment traffic, the Workstation calculates this data as follows. The inter segment traffic is derived from the node to node statistics for the intersecting set of nodes. Thus, if segment A has nodes 1, 2, and 3 and segment B has nodes 20, 21, and 22, then summing the node to node traffic for

1 -> 20,21,22
2 -> 20,21,22
3 -> 20,21,22

produces the required result. On-LAN/off-LAN traffic for segments is calculated by a simply summing node to node traffic for all stations on the LAN and then subtracting this from total segment counts.

Alarms are reported to the user in the following ways:

1. Alarms received are logged in a Workstation log.
2. The node which the alarm relates to is highlighted on the map.
3. The node status change is propagated up through the (map) hierarchy to support the case where the node is not visible on the screen. This is as provided by SNM.

Summary Tool

After the user has selected an object from the map and invokes the display tools, the summary tool generates the user's initial screen at the Management Workstation. It presents a set of statistical data selected to give an overview of the operational status of the object (e.g., a selected node or segment). The Workstation polls the Monitor for the data required by the Summary Tool display screens.

The Summary Tool displays a basic summary tool screen such as is shown in Fig. 18. The summary tool screen has three panels, namely, a control panel 602, a values panel 604, and a dialogs panel 606. The control panel includes the indicated mouse activated bottons. The functions of each of the buttons is as follows. The file button invokes a traditional file menu. The view button invokes a view menu which allows the user to modify or tailor the visual protperties of the tool. The properties button invokes a properties menu containing choices for viewing and sometimes modifying the properties of objects. The tools button invokes a tools menu which provides access to the other Workstation tools, e.g. Values Tool.

The Update Interval field allows the user to specify the frequency at which the displayed statistics are updated by polling the Monitor. The Update Once button enables the user to retrieve a single screen update. When the Update Once button is invoked not only is the screen updated but the update interval is automatically set to "none".

The type field enables the user to specify the type of network objects on which to operate, i.e., segment or node.

The name button invokes a pop up menu containing an alphabetical list of all network objects of the type selected and apply and reset buttons. The required name can then be selected from the (scrolling) list and it will be entered in the name field of the summary tool when the apply button is invoked. Alternatively, the user may enter the name directly in the summary tool name field.

The protocol button invokes a pop up menu which provides an exclusive set of protocol layers which the user may select. Selection of a layer copies the layer name into the displayed field of the summary tool when the apply operation is invoked. An example of a protocol selection menu is shown in Fig. 19. It displays the available protocols in the form of a protocol tree with multiple protocol familes. The protocol selection is two dimensional. That is, the user first selects the protocol family and then the particular layer within that family.

As indicated by the protocol trees shown in Fig. 19, the capabilities of the Monitor can be readily extended to handle other protocol families. The particular ones which are implemented depend upon the needs of the particular network environment in which the Monitor will operate.

The user invokes the apply button to indicate that the selection process is complete and the type, name, protocol, etc. should be applied. This then updates the screen using the new parameter set that the user selected. The reset button is used to undo the selections and restore them to their values at the last apply operation.

The set of statistics for the selected parameter set is displayed in values panel 604. The members of the sets differ depending upon, for example, what protocol was selected. Figs. 20a-g present examples of the types of statistical variables which are displayed for the DLL, IP, UDP, TCP, ICMP, NFS, and ARP/RARP protocols, respectively. The meaning of the values display fields are described in Appendix I, attached hereto.

Dialogs panel 606 contains a display of the connection statistics for all protocols for a selected node. Within the Management Workstation, connection lists are maintained per node, per supported protocol. When connections are displayed, they are sorted on "Last Seen" with the most current displayed first. A single list returned from the Monitor contains all current connection. For TCP, however, each connection also contains a state and TCP connections are displayed as

Past and Present based upon the returned state of the connection. For certain dialogs, such as TCP and NFS over UDP, there is an associated direction to the dialog, i.e., from the initiator (source) to the receiver (sink). For these dialogs, the direction is identified in a DIR. field. A sample of information that is displayed in dialogs panel 606 is presented in Fig. 21 for current connections.

Values Tool

The values tool provides the user with the ability to look at the statistical database for a network object in detail. When the user invokes this tool, he may select a basic data screen containing a rate values panel 620, a count values panel 622 and a protocols seen panel 626, as shown in Fig. 22, or he may select a traffic matrix screen 628, as illustrated in Fig. 23.

In rate values and count values panels 620 and 622, value tools presents the monitored rate and count statistics, respectively, for a selected protocol. The parameters which are displayed for the different protocols (i.e., different groups) are listed in Appendix II. In general, a data element that is being displayed for a node shows up in three rows, namely, a total for the data element, the number into the data element, and the number out of the data element. Any exceptions to this are identified in Appendix II. Data elements that are displayed for segments, are presented as totals only, with no distinction between Rx and Tx.

When invoked the Values Tool displays a primary screen to the user. The primary screen contains what is considered to be the most significant information for the selected object. The user can view other information for the object (i.e., the statistics for the other parameters) by scrolling down.

The displayed information for the count values and rate values panels 620 and 622 includes the following. An alarm field reports whether an alarm is currently active for this item. It displays as "*" if active alarm is present. A Current Value/Rate field reports the current rate or the value of the counter used to generate threshold alarms for this item. This is reset following each threshold trigger and thus gives an idea of how close to an alarm threshold the variable is. A Typical Value field reports what this item could be expected to read in a "normal" operating situation. This field is filled in for those items where this is predictable and useful. It is maintained in the Workstation database and is modifiable by the user using the set tool. An Accumulated Count field reports the current accumulated value of the item or the current rate. A Max Value field reports the highest value recently seen for the item. This value is reset at intervals defined by a user adjustable parameter (default 30 minutes). This is not a rolling cycle but rather represents the highest value since it was reset which may be from 1 to 30 minutes ago (for a rest period of 30 minutes). It is used only for rates. A Min Value field reports the lowest value recently seen for the item. This operates in the same manner as Max Value field and is used only for rates.

A Percent (%) field reports only for the following variables:

off seg counts:

100(in count / total off seg count)

100(out count / total off seg count)

100(transit count / total off seg count)

100(local count / total off seg count)

off seg rates

100(transit rate / total off seg rate), etc.

protocols

100(frame rate this protocol / total frame rate)

On the right half of the basic display, there the following addtional fields: a High Threshold field and a Sample period for rates field.

Set Tool

The set tool provides the user with the ability to modify the parameters controling the operation of the Monitors and the Management Workstation. These parameters affect both user interface displays and the actual operation of the Monitors. The parameters which can be operated on by the set tool can be divided into the following categories: alarm thresholds, monitoring control, segment Monitor administration, and typical values.

The monitoring control variables specify the actions of the segment Monitors and each Monitor can have a distinct set of control variables (e.g., the parse control records that are described elsewhere). The user is able to define those nodes, segments, dialogs and protocols in which he is interested so as to make the best use of memory space available for data storage. This mechanism allows for load sharing, where mulitple Monitors on the same segment can divide up the total number of network objects which are to be monitored so that no duplication of effort between them takes place.

The monitor administration variables allow the user to modify the operation of the segment Monitor in a more direct manner than the monitoring control variables. Using the set tool, the user can perform those operations such as reset, time changes etc. which are normally the prerogative of a system administrator.

Note that the above descriptions of the tools available through the Management Workstation are not meant to imply that other choices may not be made regarding the particular information which is displayed and the manner in which it is displayed.

Adaptively Setting Network Monitor Thresholds:

The Workstation sets the thresholds in the Network Monitor based upon the performance of the system as observed over an extended period of time. That is, the Workstation periodically samples the output of the Network Monitors and assembles a model of a normally functioning network. Then, the Workstation sets the thresholds in the Network Monitors based upon that model. If the observation period is chosen to be long enough and since the model represents the "average" of the network performance over the observation period, temporary undesired deviations from normal behavior are smoothed out over time and model tends to accurately reflect normal network behavior.

Referring the Fig. 24, the details of the training procedure for adaptively setting the Network Monitor thresholds are as follows. To begin training, the Workstation sends a start learning command to the Network Monitors from which performance data is desired (step 302). The start learning command disables the thresholds within the Network Monitor and causes the Network Monitor to periodically send data for a predefined set of network parameters to the Management Workstation. (Disabling the thresholds, however, is not necessary. One could have the learning mode operational in parallel with monitoring using existing thresholds.) The set of parameters may be any or all of the previously mentioned parameters for which thresholds are or may be defined.

Throughout the learning period, the Network Monitor sends "snapshots" of the network's performance to the Workstation which, in turn, stores the data in a performance history database 306 (step 304). The network manager sets the length of the learning period.

Typically, it should be long enough to include the full range of load conditions that the network experiences so that a representative performance history is generated. It should also be long enough so that short periods of overload or faulty behavior do not distort the resulting averages.

After the learning period has expired, the network manager, through the Management Workstation, sends a stop learning command to the Monitor (step 308). The Monitor ceases automatically sending further performance data updates to the Workstation and the Workstation processes the data in its performance history database (step 310). The processing may involve simply computing averages for the parameters of interest or it may involve more sophisticated statistical analysis of the data, such as computing means, standard deviations, maximum and minimum values, or using curve fitting to compute rates and other pertinent parameter values.

After the Workstation has statistically analyzed the performance data, it computes a new set of thresholds for the relevant performance parameters (step 312). To do this, it uses formulas which are appropriate to the particular parameter for which a threshold is being computed. That is, if the parameter is one for which one would expect to see wide variations in its value during network monitoring, then the threshold should be set high enough so that the normal expected variations do not trigger alarms. On the other hand, if the parameter is of a type for which only small variations are expected and larger variations indicate a problem, then the threshold should be set to a value that is close to the average observed value. Examples of formulae which may be used to compute thresholds are:

* Highest value seen during learning period;

- * Highest value seen during learning period + 10%;
- * Highest value seen during learning period + 50%;
- * Highest value seen during learning period + user-defined percent;
- * Any value of the parameter other than zero;
- * Average value seen during learning period + 50%; and
- * Average value seen during learning period + user-defined percent.

As should be evident from these examples, there is a broad range of possibilities regarding how to compute a particular threshold. The choice, however, should reflect the parameter's importance in signaling serious network problems and its normal expected behavior (as may be evidenced from the performance history acquired for the parameter during the learning mode).

After the thresholds are computed, the Workstation loads them into the Monitor and instructs the Monitor to revert to normal monitoring using the new thresholds (step 314).

This procedure provides a mechanism enabling the network manager to adaptively reset thresholds in response to changing conditions on the network, shifting usage patterns and evolving network topology. As the network changes over time, the network manager merely invokes the adaptive threshold setting feature and updates the thresholds to reflect those changes.

<u>The Diagnostic Analyzer Module</u>:

The Management Workstation includes a diagnostic analyzer module which automatically detects and diagnoses the existence and cause of certain types of network problems. The functions of the diagnostic module may actually be distributed among the Workstation and the

Network Monitors which are active on the network. In principle, the diagnostic analyzer module includes the following elements for performing its fault detection and analysis functions.

The Management Workstation contains a reference model of a normally operating network. The reference model is generated by observing the performance of the network over an extended period of time and computing averages of the performance statistics that were observed during the observation period. The reference model provides a reference against which future network performance can be compared so as to diagnose and analyze potential problems. The Network Monitor (in particular, the STATS module) includes alarm thresholds on a selected set of the parameters which it monitors. Some of those thresholds are set on parameters which tend to be indicative of the onset or the presence of particular network problems.

During monitoring, when a Monitor threshold is exceeded, thereby indicating a potential problem (e.g. in a TCP connection), the Network Monitor alerts the Workstation by sending an alarm. The Workstation notifies the user and presents the user with the option of either ignoring the alarm or invoking a diagnostic algorithm to analyze the problem. If the user invokes the diagnostic algorithm, the Workstation compares the current performance statistics to its reference model to analyze the problem and report its results. (Of course, this may also be handled automatically so as to not require user intervention.) The Workstation obtains the data on current performance of the network by retrieving the relevant performance statistics from all of the segment Network Monitors that may have information useful to diagnosing the problem.

The details of a specific example involving poor TCP connection performance will now be described. This example refers to a typical network on which the diagnostic analyzer resides, such as the network illustrated in Fig. 25. It includes three segments labelled S1, S2, and S3, a router R1 connecting S1 to S2, a router R2 connecting S2 to S3, and at least two nodes, node A on S1 which communicates with node B on S3. On each segment there is also a Network Monitor 324 to observe the performance of its segment in the manner described earlier. A Management Workstation 320 is also located on S1 and it includes a diagnostic analyzer module 322. For this example, the sympton of the network problem is degraded peformance of a TCP connection between Nodes A and B.

A TCP connection problem may manifest itself in a number of ways, including, for example, excessively high numbers for any of the following:

errors
packets with bad sequence numbers
packets retransmitted
bytes retransmitted
out of order packets
out of order bytes
packets after window closed
bytes after window closed
average and maximum round trip times

or by an unusually low value for the current window size. By setting the appropriate thresholds, the Monitor is programmed to recognize any one or more of these symptons. If any one of of the thresholds is exceeded, the Monitor sends an alarm to the Workstation. The Workstation is programmed to recognize the particular alarm as related to an event which can be further analyzed by its diagnostic analyzer module 322. Thus, the Workstation presents the user with the option of invoking its diagnostic capabilities (or automatically invokes the diagnostic capabilities).

In general terms, when the diagnostic analyzer is invoked, it looks at the performance data that the segment Monitors produce for the two nodes, for the dialogs between them and for the links that interconnect them and compares that data to the reference model for the network. If a significant divergence from the reference model is identified, the diagnostic analyzer informs the Workstation (and the user) about the nature of the divergence and the likely cause of the problem. In conducting the comparison to "normal" network performance, the network circuit involved in communications between nodes A and B is decomposed into its individual components and diagnostic analysis is performed on each link individually in the effort to isolate the problem further.

The overall structure of the diagnostic algorithm 400 is shown in Fig. 26. When invoked for analyzing a possible TCP problem between nodes A and B, diagnostic analyzer 322 checks for a TCP problem at node A when it is acting as a source node (step 402). To perform this check, diagnostic algorithm 400 invokes a source node analyzer algorithm 450 shown in Fig. 27. If a problem is identified, the Workstation reports that there is a high probability that node A is causing a TCP problem when operating as a source node and it reports the results of the investigation performed by algorithm 450 (step 404).

If node A does not appear to be experiencing a TCP problem when acting as a source node, diagnostic analyzer 322 checks for evidence of a TCP problem at node B when it is acting as a sink node (step 406). To perform this check, diagnostic algorithm 400 invokes a sink node analyzer algorithm 470 shown in Fig. 28. If a problem is identified, the Workstation reports that there is a high probability that node B is causing a TCP problem when operating as a sink node and it reports the results of the investigation performed by algorithm 470 (step 408).

Note that source and sink nodes are concepts which apply to those dialogs for which a direction of the communication can be defined. For example, the source node may be the one which initiated the dialog for the purpose of sending data to the other node, i.e., the sink node.

If node B does not appear to be experiencing a TCP problem when acting as a sink node, diagnostic analyzer 322 checks for evidence of a TCP problem on the link between Node A and Node B (step 410). To perform this check, diagnostic algorithm 400 invokes a link analysis algorithm 550 shown in Fig. 29. If a problem is identified, the Workstation reports that there is a high probability that a TCP problem exists on the link and it reports the results of the investigation performed by link analysis algorithm 550 (step 412).

If the link does not appear to be experiencing a TCP problem, diagnostic analyzer 322 checks for evidence of a TCP problem at node B when it is acting as a source node (step 414). To perform this check, diagnostic algorithm 400 invokes the previously mentioned source algorithm 450 for Node B. If a problem is identified, the Workstation reports that there is a medium probability that node B is causing a TCP problem when operating as a source node and it reports the results of the investigation performed by algorithm 450 (step 416).

If node B does not appear to be experiencing a TCP problem when acting as a source node, diagnostic analyzer 322 checks for a TCP problem at node A when it is acting as a sink node (step 418). To perform this check, diagnostic algorithm 400 invokes sink node analyzer algorithm 470 for Node A. If a problem is identified, the Network Monitor reports that there is a medium probability that node A is causing a TCP problem when operating as a sink node and it reports the results of the investigation performed by algorithm 470 (step 420).

Finally, if node A does not appear to be experiencing a TCP problem when acting as a sink node, diagnostic analyzer 322 reports that it was not able to isolate the cause of a TCP problem (step 422).

The algorithms which are called from within the above-described diagnostic algorithm will now be described. Referring to Fig. 27, source node analyzer algorithm 450 checks whether a particular node is causing a TCP problem when operating as a source node. The strategy is as follows. To determine whether a TCP problem exists at this node which is the source node for the TCP connection, look at other connections for which this node is a source. If other TCP connections are okay, then there is probably not a problem with this node. This is an easy check with a high probability of being correct. If no other good connections exist, then look at the lower layers for possible reasons. Start at DLL and work up as problems at lower layers are more fundamental, i.e., they cause problems at higher layers whereas the reverse is not true.

In accordance with this approach, algorithm 450 first determines whether the node is acting as a source node in any other TCP connection and, if so, whether the other connection is okay (step 452). If the node is performing satisfactorily as a source node in another TCP connection, algorithm 450 reports that there is no problem at the source node and returns to diagnostic algorithm 400 (step 454). If algorithm 450 cannot identify any other TCP connections involving this node that are okay, it moves up through the protocol stack checking each level for a problem. In this case, it then checks for DLL problems at the node when it is acting as a source node by calling an DLL problem checking routine 510 (see Fig. 30) (step 456). If a DLL problem is found, that fact is reported (step 458). If no DLL problems are found, algorithm 450 checks for an IP problem at the node when it is acting as a source by calling an IP problem checking routine 490 (see Fig. 31) (step 460). If an IP problem is found, that fact is reported (step 462). If no IP problems are found, algorithm 450 checks whether any other TCP connection in which the node participates as a source is not okay (step 464). If another TCP connection involving the node exists and it is not okay, algorithm 450 reports a TCP problem at the node (step 466). If no other TCP connections where the node is acting as a source node can be found, algorithm 450 exits.

Referring to Fig. 28, sink node analyzer algorithm 470 checks whether a particular node is causing a TCP problem when operating as a sink node. It first determines whether the node is acting as a sink node in any other TCP connection and, if so, whether the other connection is okay (step 472). If the node is performing satisfactorily as a sink node in another TCP connection, algorithm 470 reports that there is no problem at the source node and returns to diagnostic algorithm 400 (step 474). If algorithm 470 cannot identify any other TCP connections involving this node that are okay, it then checks for DLL problems at the node when it is acting as a sink node by calling DLL problem checking routine 510 (step 476). If a DLL problem is found, that fact is reported (step 478). If no DLL problems are found, algorithm 470 checks for an IP problem at the node when it is acting as a sink by calling IP problem checking routine 490 (step 480). If an IP problem is found, that fact is reported (step 482). If no IP problems are found, algorithm 470 checks whether any other TCP connection in which the node participates as a sink is not okay (step 484). If another TCP connection involving the node as a sink exists and it is not okay, algorithm 470 reports a TCP problem at the node (step 486). If no other TCP connections where the node is acting as a sink node can be found, algorithm 470 exits.

Referring to Fig. 31, IP problem checking routine 490 checks for IP problems at a node. It does this by comparing the IP performance statistics for the node to the reference model (steps 492 and 494). If it detects any significant deviations from the reference model, it reports that there is an IP problem at the node (step 496). If no significant deviations are noted, it reports that there is no IP problem at the node (step 498).

As revealed by examining Fig. 30, DLL problem checking routine 510 operates in a similar manner to IP problem checking routine 490, with the exception that it examines a different set of parameters (i.e., DLL parameters) for significant deviations.

Referring the Fig. 29, link analysis logic 550 first determines whether any other TCP connection for the link is operating properly (step 552). If a properly operating TCP connection exists on the link, indicating that there is no link problem, link analysis logic 550 reports that the link is okay (step 554). If a properly operating TCP connection cannot be found, the link is decomposed into its constituent components and an IP link component problem checking routine 570 (see Fig. 32) is invoked for each of the link components (step 556). IP link component problem routine 570 evaluates the link component by checking the IP layer statistics for the relevant link component.

The decomposition of the link into its components arranges them in order of their distance from the source node and the analysis of the components proceeds in that order. Thus, for example, the link components which make up the link between nodes A and B include in order: segment S1, router R1, segment S2, router R2, and segment S3. The IP data for these various components are analyzed in the following order:

IP data for segment S1
IP data for address R1
IP data for source node to R1
IP data for S1 to S2
IP data for S2
IP data for address R2
IP data for S3
IP data for S2 to S3
IP data for S1 to S3

As shown in Fig. 32, IP link component problem checking routine 570 compares IP statistics for the link component to the reference model (step 572) to determine whether network performance deviates significantly from that specified by the model (step 574). If significant deviations are detected, routine 570 reports that there is an IP problem at the link component (step 576). Otherwise, it reports that it found no IP problem (step 578).

Referring back to Fig. 29, after completing the IP problem analysis for all of the link components, logic 550 then invokes a DLL link component problem checking routine 580 (see Fig. 33) for each link component to check its DLL statistics (step 558).

DLL link problem routine 580 is similar to IP link problem routine 570. As shown in Fig. 33, DLL link problem checking routine 580 compares DLL statistics for the link to the reference model (step 582) to determine whether network performance at the DLL deviates significantly from that specified by the model (step 584). If significant deviations are detected, routine 580 reports that there is a DLL problem at the link component (step 586). Otherwise, it reports that no DLL problems were found (step 588).

Referring back to Fig. 29, after completing the DLL problem analysis for all of the link components, logic 550 checks whether there is any other TCP on the link (step 560). If another TCP exists on the link (which implies that the other TCP is also not operating properly), logic 550 reports that there is a TCP problem on the link (step 562). Otherwise, logic 550 reports that there was not enough information from the existing packet traffic to determine whether there was a link problem (step 564)

If the analysis of the link components does not isolate the source of the problem and if there were components for which sufficient information was not available (due possibly to lack of traffic over through that component), the user may send test messages to those components to generate the information needed to evaluate its performance.

The reference model against which comparisons are made to detect and isolate malfunctions may be generated by examining the behavior of the network over an extended period of operation or over multiple periods of operation. During those periods of operation, average values and maximum excursions (or standard deviations) for observed statistics are computed. These values provide an initial estimate of a model of a properly functioning system. As more experience with the network is obtained and as more historical data on the various statistics is accumulated the thresholds for detecting actual malfunctions or imminent malfunctions and the reference model can be revised to reflect the new experience.

What constitutes a significant deviation from the reference model depends upon the particular parameter involved. Some parameters will not deviate from the expected norm and thus any deviation would be considered to be significant, for example, consider ICMP messages of type "destination unreachable," IP errors, TCP errors. Other parameters will normally vary within a wide range of acceptable values, and only if they move outside of that range should the deviation be considered significant. The acceptable ranges of variation can be determined by watching network performance over a sustained period of operation.

The parameters which tend to provide useful information for identifying and isolating problems at the node level for the different protocols and layers include the following.

<u>TCP</u>
error rate
header byte rate
packets retransmitted
bytes retransmitted
packets after window closed
bytes after window closed

<u>UDP</u>
error rate
header byte rate

<u>IP</u>
error rate
header byte rate
fragmentation rate
all ICMP messages of type destination

unreachable, parameter problem,
redirection

DLL
error rate
runts

For diagnosing network segment problems, the above-identified parameters are also useful with the addition of the alignment rate and the collision rate at the DLL. All or some subset of these parameters may be included among the set of parameters which are examined during the diagnostic procedure to detect and isolate network problems.

The above-described technique can be applied to a wide range of problems on the network, including among others, the following:

TCP Connection fails to establish
UDP Connection performs poorly
UDP not working at all
IP poor performance/high error rate
IP not working at all
DLL poor performance/high error rate
DLL not working at all

For each of these problems, the diagnostic approach would be similar to that described above, using, of course, different parameters to identify the potential problem and isolate its cause.

The Event Timing Module

Referring again to Fig. 5, the RTP is programmed to detect the occurrence of certain transactions for which timing information is desired. The transactions typically occur within a dialog at a particular layer of the protocol stack and they involve a first event (i.e., an initiating event) and a subsequent partner event or response. The events are protocol messages that arrive at the Network Monitor, are parsed by the RTP and then passed to Event Timing Module (ETM) for processing. A transaction of interest might be, for example, a read of a file on a server. In that case, the initiating event is the read request and the partner event is the read response. The time of interest is the time required to receive a response to the read request (i.e., the transaction time). The transaction time provides a useful measure of network performance and if measured at various times throughout the day under different load conditions gives a measure of how different loads affect network response times. The layer of the communicaton protocol at which the relevant dialog takes place will of course depend upon the nature of the event.

In general, when the RTP detects an event, it transfers control to the ETM which records an arrival time for the event. If the event is an initiating event, the ETM stores the arrival time in an event timing database 300 (see Fig. 34) for future use. If the event is a partner event, the ETM computes a difference between that arrival time and an earlier stored time for the initiating event to determine the complete transaction time.

Event timing database 300 is an array of records 302. Each record 302 includes a dialog field 304 for identifying the dialog over which the transactions of interest are occurring and it includes an entry type field 306 for identifying the event type of interest. Each record 302 also includes a start time field 308 for storing the arrival time of the initiating event and an average delay time field 310 for storing the computed average delay for the transactions. A more detailed description of the operation of the ETM follows.

Referring to Fig. 35, when the RTP detects the arrival of a packet of the type for which timing information is being kept, it passes control to the ETM along with relevant information from the packet, such as the dialog identifier and the event type (step 320). The ETM then determines whether it is to keep timing information for that particular event by checking the event timing database (step 322). Since each event type can have multiple occurrences (i.e., there can be multiple dialogs at a given layer), the dialog identifier is used to distinguish between events of the same type for different dialogs and to identify those for which information has been requested. All of the dialog/events of interest are identified in the event timing database. If the current dialog and event appear in the event timing database, indicating that the event should be timed, the ETM determines whether the event is a starting event or an ending event so that it may be processed properly (step 324). For certain events, the absence of a start time in the entry field of the appropriate record 302 in event timing database 300 is one indicator that the event represents a start time; otherwise, it is an end time event. For other events, the ETM determines if the start time is to be set by the event type as specified in the packet being parsed. For example, if the event is a file read a start time is stored. If the event is the read completion it represents an end time. In general, each protocol event will have its own intrinsic meaning for how to determine start and end times.

Note that the arrival time is only an estimate of the actual arrival time due to possible queuing and other processing delays. Nevertheless, the delays are generally so small in comparison to the transaction times being measured that they are of little consequence.

In step 324, if the event represents a start time, the ETM gets the current time from the kernal and stores it in start time field 308 of the appropriate record in event timing database 300 (step 326). If the event represents an end time event, the ETM obtains the current time from the kernel and computes a difference between that time and the corresponding start time found in event timing database 300 (step 328). This represents the total time for the transaction of interest. It is combined with the stored average transaction time to compute a new running average transaction time for that event (step 330).

Any one of many different methods can be used to compute the running average transaction time. For example, the following formula can be used:

New Avg. = [(5 * Stored Avg.) + Transaction Time]/6.

After six transaction have been timed, the computed new average becomes a running average for the transaction times. The ETM stores this computed average in the appropriate record of event timing database 300, replacing the previous average transaction time stored in that record, and it clears start time entry field 308 for that record in preparation for timing the next transaction.

After processing the event in steps 322, 326, and 330, the ETM checks the age of all of the start time entries in the event timing database 300 to determine if any of them are too "old" (step 332). If the difference between the current time and any of the start times exceeds a preselected threshold, indicating that a partner event has not occurred within a reasonable period of time, the ETM deletes the old start time entry for that dialog/event (step 334). This insures that a missed packet for a partner event does not result in an erroneously large transaction time which throws off the running average for that event.

If the average transaction time increases beyond a preselected threshold set for timing events, an alarm is sent to the Workstation.

Two examples will now be described to illustrate the operation of the ETM for specific event types. In the first example, Node A of Fig. 25 is communicating with Node B using the NFS protocol. Node A is the client while Node B is the server. The Network Monitor resides on the same segment as node A, but this is not a requirement. When Node A issues a read request to Node B, the Network Monitor sees the request and the RTP within the Network Monitor transfers control to the ETM. Since it is a read, the ETM stores a start time in the Event Timing Database. Thus, the start time is the time at which the read was initiated.

After some delay, caused by the transmission delays of getting the read message to node B, node B performs the read and sends a response back to node A. After some further transmission delays in returning the read response, the Network Monitor receives the second packet for the event. At the time, the ETM recognizes that the event is an end time event and updates the average transaction time entry in the appropriate record with a new computed running average. The ETM then compares the average transaction time with the threshold for this event and if it has been exceeded, issues an alarm to the Workstation.

In the second example, node A is communicating with Node B using the Telnet protocol. Telnet is a virtual terminal protocol. The events of interest take place long after the initial connection has been established. Node A is typing at a standard ASCII (VT100 class) terminal which is logically (through the network) connected to Node B. Node B has an application which is receiving the characters being typed on Node A and, at appropriate times, indicated by the logic of the applications, sends characters back to the terminal located on Node A. Thus, every time node A sends characters to B, the Network Monitor sees the transmission.

In this case, there are several transaction times which could provide useful network performance information. They include, for example, the amount of time it takes to echo characters typed at the keyboard through the network and back to the display screen, the delay between typing an end of line command and seeing the completion of the application event come back or the network delays incurred in sending a packet and receiving acknowledgment for when it was received.

In this example, the particular time being measured is the time it takes for the network to send a packet and receive an acknowledgement that the packet has arrived. Since Telnet runs on top of TCP, which in turn runs on top of IP, the Network Monitor monitors the TCP acknowledge end-to-end time delays.

Note that this is a design choice of the implementation and that all events visible to the Network Monitor by virtue of the fact that information is in the packet could be measured.

When Node A transmits a data packet to Node B, the Network Monitor receives the packet. The RTP recognizes the packet as being part of a timed transaction and passes control to the ETM. The ETM recognizes it as a start time event, stores the start time in the event timing database and returns control to the RTP after checking for aging.

When Node B receives the data packet from Node A, it sends back an acknowledgment packet. When the Network Monitor sees that packet, it delivers the event to the ETM, which recognizes it as an end time event. The ETM calculates the delay time for the complete transaction and uses that to update the average transaction time. The ETM then compares the new average transaction time with the threshold for this event. If it has been exceeded, the ETM issues an alarm to the Workstation.

Note that this example is measuring something very different than the previous example. The first example measures the time it takes to traverse the network, perform an action and return that result to the requesting node. It measures performance as seen by the user and it includes delay times from the network as well as delay times from the File Server.

The second example is measuring network delays without looking at the service delays. That is, the ETM is measuring the amount of time it takes to send a packet to a node and receive the acknowledgement of the receipt of the message. In this example, the ETM is measuring transmissions delays as well as processing delays associated with network traffic, but not anything having to do with non-network processing.

As can be seen from the above examples, the ETM can measure a broad range of events. Each of these events can be measured passively and without the cooperation of the nodes that are actually participating in the transmission.

The Address Tracker Module (ATM)

Address tracker module (ATM) 43, one of the software modules in the Network Monitor (see Fig. 5), operates on networks on which the node addresses for particular node to node connections are assigned dynamically. An Appletalk® Network, developed by Apple Computer Company, is an example of a network which uses dynamic node addressing. In such networks, the dynamic change in the address of a particular service causes difficulty troubleshooting the network because the network manager may not know where the various nodes are and what they are called. In addition, foreign network addresses (e.g., the IP addresses used by that node for communication over an IP network to which if is connected) can not be relied upon to point to a particular node. ATM 43 solves this problem by passively monitoring the network traffic and collecting a table showing the node address to node name mappings.

In the following description, the network on which the Monitor is located is assumed to be an Appletalk® Network. Thus, as background for the following discussion, the manner in which the dynamic node addressing mechanism operates on that network will first be described.

When a node is activated on the Appletalk® Network, it establishes its own node address in accordance with protocol referred to as the Local Link Access Protocol (LLAP). That is, the node guesses its own node address and then verifies that no other node on the network is using that address. The node verifies the uniqueness of its guess by sending an LLAP Enquiry control packet informing all other nodes on the network that it is going to assign itself a particular address unless another node responds that the address has already been assigned. If no other node claims that address as its own by sending an LLAP acknowledgment control packet, the first node uses the address which it has selected. If another node claims the address as its own, the first node tries another address. This continues until, the node finds an unused address.

When the first node wants to communicate with a second node, it must determine the dynamically assigned node address of the second node. It does this in accordance with another protocol referred to as the Name Binding Protocol (NBP). The Name Binding Protocol is used to map or bind human understandable node names with machine understandable node addresses. The NBP allows nodes to dynamically translate a string of characters (i.e., a node name) into a node address. The node needing to communicate with another node broadcasts an NBP Lookup packet containing the name for which a node address is being requested. The node having the name being requested responds with its address and returns a Lookup Reply packet containing its address to the original requesting node. The first node then uses that address its current communications with the second node.

Referring to Fig. 36, the network includes an Appletalk® Network segment 702 and a TCP/IP segment 704, each of which are connected to a larger network 706 through their respective gateways 708. A Monitor 710, including a Real Time Parser (RTP) 712 and an Address Tracking Module (ATM) 714, is located on Appletalk network segment 702 along with other nodes 711. A Management Workstation 716 is located on segment 704. It is assumed that Monitor 710 has the features and capabilities previously described; therefore, those features not specifically related to the dynamic node addressing capability will not be repeated here but rather the reader is referred to the earlier discussion. Suffice it to say that Monitor 710 is, of course, adapted to operate on Appletalk Network segment 702, to parse and analyze the packets which are transmitted over that segment according to the Appletalk® family of protocols and to communicate the information which it extracts from the network to Management Workstation 716 located on segment 704.

Within Monitor 710, ATM 714 maintains a name table data structure 730 such as is shown in Fig. 37. Name Table 720 includes records 722, each of which has a node name field 724, a node address field 726, an IP address field 728, and a time field 729. ATM 714 uses Name Table 720 to keep track of the mappings of node names to node address and to IP address. The relevance of each of the fields of records 722 in Name Table 720 are explained in the following description of how ATM 714 operates.

In general, Monitor 710 operates as previously described. That is, it passively monitors all packet traffic over segment 702 and sends all packets to RTP 712 for parsing. When RTP 712 recognizes an Appletalk packet, it transfers control to ATM 714 which analyzes the packet for the presence of address mapping information.

The operation of ATM 714 is shown in greater detail in the flow diagram of Fig. 38. When ATM 714 receives control from RTP 712, it takes the packet (step 730 and strips off the lower layers of the protocol until it determines whether there is a Name Binding Protocol message inside the packet (step 732). If it is a NBP message, ATM 714 then determines whether it is new name Lookup message (step 734). If it is a new name Lookup message, ATM 714 extracts the name from the message (i.e., the name for which a node address is being requested) and adds the name to the node name field 724 of a record 722 in Name Table 720 (step 736).

If the message is an NBP message but it is not a Lookup message, ATM 714 determines whether it is a Lookup Reply (step 738). If it is a Lookup Reply, signifying that it contains a node name/node address binding, ATM 714 extracts the name and the assigned node address from the message and adds this information to Name Table 720. ATM 714 does this by searching the name fields of records 722 in Name Table 720 until it locates the name. Then, it updates the node address field of the identified record to contain the node address which was extracted from the received NBP packet. ATM 714 also updates time field 729 to record the time at which the message was processed.

After ATM 714 has updated the address field of the appropriate record, it determines whether any records 722 in Name Table 720 should be aged out (step 742). ATM 714 compares the current time to the times recorded in the time fields. If the elapsed time is greater than a preselected time period (e.g. 48 hours), ATM 714 clears the record of all information (step 744). After that, it awaits the next packet from RTP 712.

As ATM 714 is processing each a packet and it determines either that it does not contain an NBP message (step 732) or it does not contain a Lookup Reply message (step 738), ATM 714 branches to step 742 to perform the age out check before going on to the next packet from RTP 712.

The Appletalk to IP gateways provide services that allow an Appletalk Node to dynamically connect to an IP address for communicating with IP nodes. This service extends the dynamic node address mechanism to the IP world for all Appletalk nodes. While the flexibility provided is helpful to the users, the network manager is faced with the problem of not knowing which Appletalk Nodes are currently using a particular IP address and thus, they can not easily track down problems created by the particular node.

ATM 714 can use passive monitoring of the IP address assignment mechanisms to provide the network manager a Name-to-IP address mapping.

If ATM 714 is also keeping IP address information, it implements the additional steps shown in Fig. 39 after completing the node name to node address mapping steps. ATM 714 again checks whether it is an NBP message (step 748). If it is an NBP message, ATM 714 checks whether it is a response to an IP address request (step 750). IP address requests are typically implied by an NBP Lookup request for an IP gateway. The gateway responds by supplying the gateway address as well as an IP address that is assigned to the requesting node. If the NBP message is an IP address response, ATM 714 looks up the requesting node in Name Table 720 (step 752) and stores the IP address assignment in the IP address field of the appropriate record 722 (step 754).

After storing the IP address assignment information, ATM 714 locates all other records 722 in Name Table 720 which contain that IP address. Since the IP address has been assigned to a new node name, those old entries are no longer valid and must be eliminated. Therefore, ATM 714 purges the IP address fields of those records (step 756). After doing this cleanup step, ATM 714 returns control to RTP 712.

Other embodiments are within the following claims. For example, the Network Monitor can be adapted to identify node types by analyzing the type of packet traffic to or from the node. If the node being monitored is receiving mount requests, the Monitor would report that the node is behaving like node a file server. If the node is issuing routing requests, the Monitor would report that the node is behaving like a router. In either case, the network manager can check a table of what nodes are permitted to provide what functions to determine whether the node is authorized to function as either a file server or a router, and if not, can take appropriate action to correct the problem.

APPENDIX I

SNMP MIB Subset Supported

This is the subset of the standard MIB which can be obtained by monitoring.

Refer to RFC 1066 Management Information Base for an explanation on the items which follow.

System group:
none

Interfaces group
ifType
ifPhysAddress
ifOperStatus
ifInOctets
ifInUcastPkts
ifInNUcastPkts
ifOutOctets
ifOutUcastPkts
ifOutNUcastPkts

Address Translation group
none

IP group
ipForwarding
ipDefaultTTL
ipInReceives
ipInHdrErrors
ipInAddrErrors
ipForwDatagrams
ipReasmReqds
ipFragCreates

IP Address Table
ipAddress
ipAdEntBcastAddr

IP Routing Table
none

ICMP group
icmpInMsgs
icmplnErrors
icmpInDestUnreachs
icmpInTimeExcds
icmpInParmProbs
icmpInSrcQuenchs
icmpInRedirects
icmpInEchoes

XP2031754

Second Edition

Internetworking with TCP/IP

VOLUME I

Principles, Protocols, and Architecture

Douglas E. Comer

Contents

Chapter 6 Determining an Internet Address at Startup (RARP) 83

Chapter 7 Internet Protocol: Connectionless Datagram Delivery 89

Chapter 8 Internet Protocol: Routing IP Datagrams 109

Chapter 13 Routing: Cores, Peers, and Algorithms (GGP) 205

Chapter 14 Routing: Autonomous Systems (EGP) 223

Chapter 15 Routing: Interior Gateway Protocols (RIP, OSPF, HELLO) 243

Chapter 21 The Socket Interface 335

Chapter 22 Applications: Remote Login (TELNET, Rlogin) 365

1

Introduction and Overview

1.1 The Need For An Internet

Data communication has become a fundamental part of computing. World-wide networks gather data about such diverse subjects as atmospheric conditions, crop production, and airline traffic. Groups establish electronic mailing lists so they can share information of common interest. Hobbyists exchange programs for their home computers. In the scientific world, data networks are essential because they allow scientists to send programs and data to remote supercomputers for processing, to retrieve the results, and to exchange scientific information with colleagues.

Unfortunately, most networks are independent entities, established to serve the needs of a single group. The users choose a hardware technology appropriate to their communication problems. More important, it is impossible to build a universal network from a single hardware technology because no single network suffices for all uses. Some users need a high-speed network to connect machines, but such networks cannot be expanded to span large distances. Others settle for a slower speed network that connects machines thousands of miles apart.

Recently, however, a new technology has emerged that makes it possible to interconnect many disparate physical networks and make them function as a coordinated unit. The new technology, called *internetworking*, or *internetting*, accommodates multiple, diverse underlying hardware technologies by adding both physical connections and a new set of conventions. The internet technology hides the details of network hardware and permits computers to communicate independent of their physical network connections.

The internet technology described in this book is an example of *open system interconnection*. It is called an *open system* because, unlike proprietary communication systems available from one specific vendor, the specifications are publicly available. Thus, anyone can build the software needed to communicate across an internet. More important, the entire technology has been designed to foster communication between machines with diverse hardware architectures, to use almost any packet switched network hardware, and to accommodate multiple computer operating systems.

To appreciate internet technology, think of how it affects research. Imagine for a minute the effects of interconnecting all the computers used by scientists. Any scientist would be able to exchange data resulting from an experiment with any other scientist. It would be possible to establish national data centers to collect data from natural phenomena and make the data available to all scientists. Computer services and programs available at one location could be used by scientists at other locations. As a result, the speed with which scientific investigations proceed would increase. In short, the changes would be dramatic.

1.2 The TCP/IP Internet

Government agencies have realized the importance and potential of internet technology for many years and have been funding research that will make possible a national internet. This book discusses principles and ideas underlying the leading internet technology, one that has resulted from research funded by the *Defense Advanced Research Projects Agency* (*DARPA*). The DARPA technology includes a set of network standards that specify the details of how computers communicate, as well as a set of conventions for interconnecting networks and routing traffic. Officially named the TCP/IP Internet Protocol Suite and commonly referred to as *TCP/IP* (after the names of its two main standards), it can be used to communicate across any set of interconnected networks. For example, some corporations use TCP/IP to interconnect all networks within their corporation, even though the corporation has no connection to outside networks. Other groups use TCP/IP for long haul communication among geographically distant sites.

Although the TCP/IP technology is noteworthy by itself, it is especially interesting because its viability has been demonstrated on a large scale. It forms the base technology for a large internet that connects most major research institutions, including university, corporate, and government labs. The *National Science Foundation* (*NSF*), the *Department of Energy* (*DOE*), the *Department of Defense* (*DOD*), the *Health and Human Services Agency*, (*HHS*) and the *National Aeronautics and Space Administration* (*NASA*) all participate, using TCP/IP to connect many of their research sites with those of DARPA. The resulting entity, known as the *connected Internet*, the *DARPA/NSF Internet*, the *TCP/IP Internet*, or just the *Internet*†, allows researchers at connected institutions to share information with colleagues across the country as easily as they share it with researchers in the next room. An outstanding success, the Internet demonstrates the viability of the TCP/IP technology and shows how it can accommodate a wide variety of underlying network technologies.

†We will follow the usual convention of capitalizing *Internet* when referring specifically to the connected internet, and use lower case otherwise; we will also assume the term "internet" used without further qualification refers to TCP/IP internets.

Most of the material in this book applies to any internet that uses TCP/IP, but some chapters refer specifically to the connected Internet. Readers interested only in the technology should be careful to watch for the distinction between the Internet architecture as it exists and general TCP/IP internets as they might exist. It would be a mistake, however, to ignore sections of the text that describe the connected Internet completely – many corporate networks are already more complex than the connected Internet of ten years ago, and many of the problems they face have already been solved in the connected Internet.

1.3 Internet Services

One cannot appreciate the technical details underlying TCP/IP without understanding the services it provides. This chapter reviews internet services briefly, highlighting the services most users access, and leaving to later chapters the discussion of how computers connect to a TCP/IP internet and how the functionality is implemented.

Much of our discussion of services will focus on standards called *protocols*. Protocols, like TCP and IP, give the formulas for passing messages, specify the details of message formats, and describe how to handle error conditions. Most important, they allow us to discuss communication standards independent of any particular vendor's network hardware. In a sense, protocols are to communication what programming languages are to computation. A programming language allows one to specify or understand a computation without knowing the details of any particular CPU instruction set. Similarly, a communication protocol allows one to specify or understand data communication without depending on detailed knowledge of a particular vendor's network hardware.

Hiding the low-level details of communication helps improve productivity in several ways. First, because programmers deal with higher-level protocol abstractions, they do not need to learn or remember as many details about a given hardware configuration. They can create new programs quickly. Second, because programs built using higher-level abstractions are not restricted to a particular machine architecture or particular network hardware, they do not need to be changed when machines or networks are reconfigured. Third, because application programs built using higher-level protocols are independent of the underlying hardware, they can provide direct communication for an arbitrary pair of machines. Programmers do not need to build special versions of application software to move and translate data between each possible pair of machine types.

We will see that all network services are described by protocols. The next sections refer to protocols used to specify application-level services as well as those used to define network-level services. Later chapters explain each of these protocols in more detail.

2

Review of Underlying Network Technologies

2.1 Introduction

It is important to understand that the Internet is not a new kind of physical network. It is, instead, a method of interconnecting physical networks and a set of conventions for using networks that allow the computers they reach to interact. While hardware technology plays only a minor role in the overall design, it is important to be able to distinguish between the low-level mechanisms provided by the hardware itself and the higher-level facilities that the Internet protocol software provides. It is also important to understand how the facilities supplied by packet-switched technology affect our choice of high-level abstractions.

This chapter introduces basic packet-switching concepts and terminology and then reviews some of the underlying network hardware technologies that have been used in TCP/IP internets. Later chapters describe how these networks are interconnected and how the TCP/IP protocols accommodate vast differences in the hardware. While the list presented here is certainly not comprehensive, it clearly demonstrates the variety among physical networks over which TCP/IP operates. The reader can safely skip many of the technical details, but should try to grasp the idea of packet switching and try to imagine building a homogeneous communication system using such heterogeneous hardware. Most important, the reader should look closely at the details of the physical address schemes the various technologies use; later chapters will discuss in detail how high-level protocols use these physical addresses.

2.2 Two Approaches To Network Communication

Whether they provide connections between one computer and another or between terminals and computers, communication networks can be divided into two basic types: *circuit-switched* and *packet-switched*†. Circuit-switched networks operate by forming a dedicated connection (circuit) between two points. The U.S. telephone system uses circuit switching technology – a telephone call establishes a circuit from the originating phone through the local switching office, across trunk lines, to a remote switching office, and finally to the destination telephone. While a circuit is in place, the phone equipment samples the microphone repeatedly, encodes the samples digitally, and transmits them across the circuit to the receiver. The sender is guaranteed that the samples can be delivered and reproduced because the circuit provides a guaranteed data path of 64 Kbps (thousand bits per second), the rate needed to send digitized voice. The advantage of circuit switching lies in its guaranteed capacity: once a circuit is established, no other network activity will decrease the capacity of the circuit. One disadvantage of circuit switching is cost: circuit costs are fixed, independent of traffic. For example, one pays a fixed rate when making a phone call, even when the two parties do not talk.

Packet-switched networks, the type usually used to connect computers, take an entirely different approach. In a packet-switched network, traffic on the network is divided into small pieces called *packets* that are multiplexed onto high capacity intermachine connections. A packet, which usually contains only a few hundred bytes of data, carries identification that enables computers on the network to know whether it is destined for them or how to send it on to its correct destination. For example, a file to be transmitted between two machines may be broken into many packets that are sent across the network one at a time. The network hardware delivers the packets to the specified destination, where network software reassembles them into a single file again. The chief advantage of packet-switching is that multiple communications among computers can proceed concurrently, with intermachine connections shared by all pairs of machines that are communicating. The disadvantage, of course, is that as activity increases, a given pair of communicating computers receives less of the network capacity. That is, whenever a packet switched network becomes overloaded, computers using the network must wait before they can send additional packets.

Despite the potential drawback of not being able to guarantee network capacity, packet-switched networks have become extremely popular. The motivations for adopting packet switching are cost and performance. Because multiple machines can share a network, fewer interconnections are required and cost is kept low. Because engineers have been able to build high speed network hardware, capacity is not usually a problem. So many computer interconnections use packet-switching that, throughout the remainder of this text, the term *network* will refer only to packet-switched networks.

†In fact, it is possible to build hybrid hardware technologies; for our purposes, only the difference i functionality is important.

2.3 Wide Area, Metropolitan Area, and Local Area Networks

Packet-switched networks that span large geographical distances (e.g., the continental U.S.) are fundamentally different from those that span short distances (e.g., a single room). To help characterize the differences in capacity and intended use, packet switched technologies are often divided into three broad categories: *wide area networks* (*WAN*s), *Metropolitan Area Networks* (*MAN*s), and *Local Area Networks* (*LAN*s).

WAN technologies, sometimes called *long haul networks*, allow endpoints to be arbitrarily far apart and are intended for use over large distances. Usually, WANs operate at slower speeds than other technologies and have much greater delay between connections. Typical speeds for a WAN range from 9.6 Kbps to 45 Mbps (million bits per second).

The newest type of network hardware, MAN technologies span intermediate geographic areas and operate at medium-to-high speeds. The name is derived from the ability of a single MAN to span a large metropolitan area. MANs introduce less delay than WANs, but cannot span as large a distance. Typical MANs operate at 56 Kbps to 100 Mbps.

LAN technologies provide the highest speed connections among computers, but sacrifice the ability to span large distances. For example, a typical LAN spans a small area like a single building or a small campus and operates between 4 Mbps and 2 Gbps (billion bits per second).

We have already mentioned the general tradeoff between speed and distance: technologies that provide higher speed communication operate over shorter distances. There are other differences among technologies in the three categories as well. In LAN technologies, each computer usually contains a network interface device that connects the machine directly to the network medium (e.g., a passive copper wire or coaxial cable). Often, the network itself is passive, depending on electronic devices in the attached computers to generate and receive the necessary electrical signals. In MAN technologies, a network contains active switching elements that introduce short delays as they route data to its destination. In WAN technologies, a network usually consists of a series of complex packet switches interconnected by communication lines. The size of the network can be extended by adding a new switch and another communication line. Attaching a computer to a WAN means connecting it to one of the packet switches. The switches introduce significant delays when routing traffic. Thus, the larger the WAN becomes the longer it takes to route traffic across it.

The goal of network protocol design is to hide the technological differences between networks, making interconnection independent of the underlying hardware. The next sections present six examples of network technologies used throughout the Internet, showing some of the differences among them. Later chapters show how the TCP/IP software isolates such differences and makes the communication system independent of the underlying hardware technology.

20

The Domain Name System

20.1 Introduction

So far we have used 32-bit integers called Internet Protocol addresses (IP addresses) to identify machines. Although such addresses provide a convenient, compact representation for specifying the source and destination in packets sent across an internet, users prefer to assign machines pronounceable, easily remembered names.

This chapter considers a scheme for assigning meaningful high-level names to a large set of machines, and it discusses a mechanism that maps between high-level machine names and IP addresses. It considers both the translation from high-level names to IP addresses and the translation from IP addresses to high-level machine names. The naming scheme is interesting for two reasons. First, it has been used to assign machine names throughout the connected Internet. Second, the implementation of the name mapping mechanism provides a large scale example of the client–server paradigm described in Chapter 18, because it uses a geographically distributed set of servers to map names to addresses.

20.2 Names For Machines

The earliest computer systems forced users to understand numeric addresses for objects like system tables and peripheral devices. Timesharing systems advanced computing by allowing users to invent meaningful symbolic names for both physical objects (e.g., peripheral devices) and abstract objects (e.g., files). A similar pattern has emerged in computer networking. Early systems supported point-to-point connections between computers and used low-level hardware addresses to specify machines. Internetworking introduced universal addressing, as well as protocol software to map universal addresses into low-level hardware addresses. Users whose computing environment contains multiple machines want meaningful, symbolic names to identify them.

Early machine names reflected the small environment in which they were chosen. It was quite common for a site with a handful of machines to choose names based on the machines' purposes. For example, machines often had names like *research, production, accounting*, and *development*. Users find such names preferable to cumbersome hardware addresses.

Although the distinction between *address* and *name* is intuitively appealing, it is artificial. Any *name* is merely an identifier that consists of a sequence of characters chosen from a finite alphabet. Names are only useful if the system can efficiently map them to the object they denote. Thus, we think of an IP address as a *low-level name*, and we say that users prefer *high-level names* for machines.

The form of high-level names is important because it determines how names are translated to lower-level names or bound to objects, as well as how name assignments are authorized. When only a few machines interconnect, choosing names is easy, and any form will suffice. On the Internet, to which over one hundred thousand machines connect, choosing symbolic names becomes difficult. For example, when the Computer Science Department at Purdue University connected to the Internet in 1980, it chose the name *purdue* to identify the connected machine. The list of potential conflicts contained only a few dozen names. By mid 1986, the official list of hosts on the Internet contained 3100 officially registered names and 6500 official aliases. By 1990, the list contained nearly 6400 names†. Although the list was growing rapidly, most sites had additional machines (e.g., personal computers) that were not registered.

20.3 Flat Namespace

The original set of machine names used throughout the Internet formed a *flat namespace* in which each name consisted of a sequence of characters without any further structure. In the original scheme, a central site, the Internet Network Information Center (NIC), administered the namespace and determined whether a new name was appropriate (i.e., it prohibited obscene names or names that conflicted with existing ones).

The chief advantage of a flat namespace is that names are convenient and short; the chief disadvantage of a flat namespace is that it cannot generalize to large sets of machines for both technical and administrative reasons. First, because names are drawn from a single set of identifiers, the potential for conflict increases as the number of sites increases. Second, because authority for adding new names must rest at a single site, the administrative workload at that central site also increases with the number of sites. To understand the severity of the problem, imagine a rapidly growing internet with thousands of sites, each of which has hundreds of individual personal computers and workstations. Every time someone acquires and connects a new personal computer, its name must be approved by the central authority. Third, because the name-to-address bindings change frequently, the cost of maintaining correct copies of the entire list at each site is high and increases as the number of sites increases. Alternatively, if the

†In 1990, the list of names maintained by the NIC was obsolete: at that time the Internet domain name system contained more than 137,000 host names.

name database resides at a single site, traffic to that site increases with the number of sites.

20.4 Hierarchical Names

How can a naming system accommodate a large, rapidly expanding set of names without requiring a central site to administer it? The answer lies in decentralizing the naming mechanism by delegating authority for parts of the namespace and distributing responsibility for the mapping between names and addresses. TCP/IP internets now use such a scheme. Before examining the details of the TCP/IP scheme, we will consider the motivation and intuition behind it.

The partitioning of a namespace must be defined in such a way that it supports efficient name mapping and guarantees autonomous control of name assignment. Optimizing only for efficient mapping can lead to solutions that retain a flat namespace and reduce traffic by dividing the names among multiple mapping machines. Optimizing only for administrative ease can lead to solutions that make delegation of authority easy but name mapping expensive or complex.

To understand how the namespace should be divided, think of the internal structure of large organizations. At the top, the chief executive has overall responsibility. Because the chief executive cannot oversee everything, the organization may be partitioned into divisions, with an executive in charge of each division. The chief executive grants each division autonomy within specified limits. More to the point, the executive in charge of a particular division can hire or fire employees, assign offices, and delegate authority, without obtaining direct permission from the chief executive.

Besides making it easy to delegate authority, the hierarchy of a large organization introduces autonomous operation. For example, when office workers need information like telephone numbers of new employees, they begin by asking local clerical workers (who may contact clerical workers in other divisions). The point is that although authority always passes down the corporate hierarchy, information can flow across the hierarchy from one office to another.

20.5 Delegation Of Authority For Names

A hierarchical naming scheme works like the management of a large organization. The namespace is *partitioned* at the top level, and authority for names in the subdivisions is passed to a designated agent. For example, we might choose to partition the namespace based on *site name* and to delegate to each site responsibility for maintaining names within its partition. The topmost level of the hierarchy divides the namespace and delegates authority for each division; it need not be bothered by changes within one division.

The syntax of hierarchically assigned names often reflects the hierarchical delegation of authority used to assign them. As an example, consider a namespace with names of the form:

local . *site*

where *site* is the site name authorized by the central authority, *local* is the part of a name controlled by the site, and the period† (".") is a delimiter used to separate them. When the topmost authority approves adding a new site, *X*, it adds *X* to the list of valid sites and delegates to site *X* authority for all names that end in '.*X*'.

20.6 Subset Authority

In a hierarchical namespace, authority may be further subdivided at each level. In our example of partition by sites, the site itself may consist of several administrative groups, and the site authority may choose to subdivide its namespace among the groups. The idea is to keep subdividing the namespace until each subdivision is small enough to be manageable.

Syntactically, subdividing the namespace introduces another partition of the name. For example, adding a *group* subdivision to names already partitioned by site produces the following name syntax:

local . *group* . *site*

Because the topmost level delegates authority, group names do not have to agree among sites. A university site might choose group names like *engineering*, *science*, and *arts*, while a corporate site might choose group names like *production*, *accounting*, and *personnel*.

The U.S. Telephone system provides another example of a hierarchical naming syntax. The 10 digits of a phone number have been partitioned into a 3-digit *area code*, 3-digit *exchange*, and 4-digit *subscriber number* within the exchange. Each exchange has authority for assigning subscriber numbers within its piece of the namespace. Although it is possible to group arbitrary subscribers into exchanges and to group arbitrary exchanges into area codes, the assignment of telephone numbers is not capricious; they are carefully chosen to make it easy to route phone calls across the telephone network.

The telephone example is important because it illustrates a key distinction between the hierarchical naming scheme used in a TCP/IP internet and other hierarchies: partitioning the set of machines owned by an organization along lines of authority does not necessarily imply partitioning by physical location. For example, it could be that at some university, a single building houses the mathematics department, as well as the computer science department. It might even turn out that although the machines from these two groups fall under completely separate administrative domains, they connect to the same physical network. It also may happen that a single group owns machines on

†In domain names, the period delimiter is pronounced "dot."

several physical networks. For these reasons, the TCP/IP naming scheme allows arbitrary delegation of authority for the hierarchical namespace without regard to physical connections. The concept can be summarized:

> *In a TCP/IP internet, hierarchical machine names are assigned according to the structure of organizations that obtain authority for parts of the namespace, not necessarily according to the structure of the physical network interconnections.*

Of course, at many sites the organizational hierarchy corresponds with the structure of physical network interconnections. At a large university, for example, most departments that have computers also have their own local area network. If the department is assigned part of the naming hierarchy, all machines that have names in its part of the hierarchy will also connect to a single physical network.

20.7 TCP/IP Internet Domain Names

The mechanism that implements a machine name hierarchy for TCP/IP internets is called the *domain name system (DNS)*. It has two, conceptually independent, aspects. The first is abstract: it specifies the name syntax and rules for delegating authority over names. The second is concrete: it specifies the implementation of a distributed computing system that efficiently maps names to addresses. This section considers the name syntax and later sections examine the implementation.

The domain name system uses a hierarchical naming scheme known as *domain names*. As in our earlier examples, a domain name consists of a sequence of subnames separated by a delimiter character, the period. In our examples we said that individual sections of the name might represent sites or groups, but the domain system simply calls each section a *label*. Thus, the domain name

cs.purdue.edu

contains three *labels*: *cs*, *purdue*, and *edu*. Any suffix of a label in a domain name is also called a *domain*. In the above example the lowest level domain is *cs.purdue.edu*, (the domain name for the Computer Science Department at Purdue University), the second level domain is *purdue.edu* (the domain name for Purdue University), and the top-level domain is *edu* (the domain name for educational institutions). As the example shows, domain names are written with the local label first and the top domain last. As we will see, writing them in this order makes it possible to compress messages that contain multiple domain names.

20.8 Official And Unofficial Internet Domain Names

In theory, the domain name standard specifies an abstract hierarchical namespace with arbitrary values for labels. Because the domain system dictates only the form of names and not their actual values, it is possible for any group that builds an instance of the domain system to choose labels for all parts of its hierarchy. For example, a private company can establish a domain hierarchy in which the top-level labels specify corporate subsidiaries, the next level labels specify corporate divisions, and the lowest level labels specify departments.

However, most users of the domain technology follow the hierarchical labels used by the official Internet domain system. There are two reasons. First, as we will see, the Internet scheme is both comprehensive and flexible. It can accommodate a wide variety of organizations, and allows each group to choose between geographical or organizational naming hierarchies. Second, most sites follow the Internet scheme so they can attach their TCP/IP installations to the connected Internet without changing names. Because the Internet naming scheme dominates almost all uses of the domain name system, examples throughout the remainder of this chapter have labels taken from the Internet naming hierarchy. Readers should remember that, although they are most likely to encounter these particular labels, the domain name system technology can be used with other labels if desired.

The Internet authority has chosen to partition its top level into the domains listed in Figure 20.1.

Domain Name	Meaning
COM	Commercial organizations
EDU	Educational institutions
GOV	Government institutions
MIL	Military groups
NET	Major network support centers
ORG	Organizations other than those above
ARPA	Temporary ARPANET domain (obsolete)
INT	International organizations
country code	Each country (geographic scheme)

Figure 20.1 The top-level Internet domains and their meanings. Although labels are shown in upper case, domain name system comparisons are insensitive to case, so *EDU* is equivalent to *edu*.

Conceptually, the top-level names permit two completely different naming hierarchies: geographic and organizational. The geographic scheme divides the universe of machines by country. Machines in the United States fall under the top-level domain *US*; when foreign countries want to register machines in the domain name system, the central authority assigns the country a new top-level domain with the country's international standard 2-letter identifier as its label. The authority for the US domain has chosen to divide it into one second-level domain per state. For example, the domain for the state of Virginia is

va . us

As an alternative to the geographic hierarchy, the top-level domains also allow organizations to be grouped by organizational type. When an organization wants to participate in the domain naming system, it chooses how it wishes to be registered and requests approval. The central authority reviews the application and assigns the organization a subdomain† under one of the existing top-level domains. For example, it is possible for a university to register itself as a second-level domain under *EDU* (the usual practice), or to register itself under the state and country in which it is located. So far, few organizations have chosen the geographic hierarchy; most prefer to register under *COM*, *EDU*, *MIL*, or *GOV*. There are two reasons. First, geographic names are longer and therefore more difficult to type. Second, geographic names are much more difficult to discover or guess. For example, Purdue University is located in West Lafayette, Indiana. While a user could easily guess an organizational name, like *purdue.edu*, a geographic name is often difficult to guess because it is usually an abbreviation, like *laf . in . us*.

Figure 20.2 illustrates a small part of the Internet domain name hierarchy. As the figure shows, Digital Equipment Corporation, a commercial organization, registered as *dec . com*. Purdue University registered as *purdue . edu*, and the National Science Foundation, a government agency, registered as *nsf . gov*. In contrast, the Corporation for National Research Initiatives chose to register under the geographic hierarchy as *nri . reston . va . us*.

†The standard does not define the term "subdomain." We have chosen to use it because its analogy to "subset" helps clarify the relationship among domains.

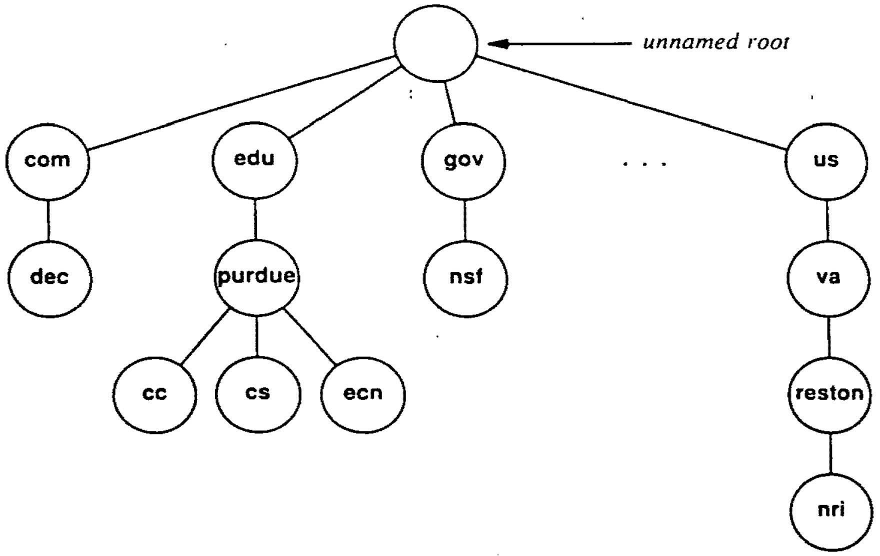


Figure 20.2 A small part of the Internet domain name hierarchy (tree). In practice, the tree is broad and flat: over one hundred thousand host entries appear by the fifth level.

Another example may help clarify the relationship between the naming hierarchy and authority for names. A machine named *xinu* in the Computer Science Department at Purdue University has the official domain name

xinu . cs . purdue . edu

The machine name was approved and registered by the local network manager in the Computer Science Department. The department manager had previously obtained authority for the subdomain *cs . purdue . edu* from a university network authority, who had obtained permission to manage the subdomain *purdue . edu* from the Internet authority. The Internet authority retains control of the *edu* domain, so new universities can only be added with its permission. Similarly, the university network manager at Purdue University retains authority for the *purdue . edu* subdomain, so new third-level domains may only be added with the manager's permission.

20.9 Items Named And Syntax Of Names

The domain name system is quite general because it allows multiple naming hierarchies to be embedded in one system. To allow clients to distinguish among multiple kinds of entries, each named item stored in the system is assigned a *type* that specifies whether it is the address of a machine, a mailbox, a user, and so on. When a client asks the domain system to resolve a name, it must specify the type of answer desired. For example, when an electronic mail application uses the domain system to resolve a name, it specifies that the answer should be the address of a *mail exchanger*. A remote login application specifies that it seeks a machine's IP address. It is important to understand the following:

> *A given name may map to more than one item in the domain system. The client specifies the type of object desired when resolving a name, and the server returns objects of that type.*

In addition to specifying the type of answer sought, the domain system allows the client to specify the protocol family to use. The domain system partitions the entire set of names by *class*, allowing a single database to store mappings for multiple protocol suites†.

The syntax of a name does not determine what type of object it names or the class of protocol suite. In particular, the number of labels in a name does not determine whether the name refers to an individual object (machine) or a domain. Thus, in our example, it is possible to have a machine named

gwen.purdue.edu

even though

cs.purdue.edu

names a subdomain. We can summarize this important point:

> *One cannot distinguish the names of subdomains from the names of individual objects or the type of an object using only the domain name syntax.*

20.10 Mapping Domain Names To Addresses

In addition to the rules for name syntax and delegation of authority, the domain name scheme includes an efficient, reliable, general purpose, distributed system for mapping names to addresses. The system is distributed in the technical sense, meaning that a set of servers operating at multiple sites cooperatively solve the mapping problem. It is efficient in the sense that most names can be mapped locally; only a few re-

†Currently, few domain servers use multiple protocol suites.

quire internet traffic. It is general purpose because it is not restricted to machine names (although we will use that example for now). Finally, it is reliable in that no single machine failure will prevent the system from operating correctly.

The domain mechanism for mapping names to addresses consists of independent. cooperative systems called *name servers*. A name server is a server program that supplies name-to-address translation, mapping from domain names to IP addresses. Often. server software executes on a dedicated processor, and the machine itself is called the name server. The client software, called a *name resolver*, uses one or more name servers when translating a name.

The easiest way to understand how domain servers work is to imagine them arranged in a tree structure that corresponds to the naming hierarchy, as Figure 20.3 illustrates. The root of the tree is a server that recognizes the top-level domains and knows which server resolves each domain. Given a name to resolve, the root can choose the correct server for that name. At the next level, a set of name servers each provide answers for one top-level domain (e.g., *edu*). A server at this level knows which servers can resolve each of the subdomains under its domain. At the third level of the tree, name servers provide answers for subdomains (e.g., *purdue* under *edu*). The conceptual tree continues with one server at each level for which a subdomain has been defined.

Links in the conceptual tree do not indicate physical network connections. Instead. they show which other name servers a given server knows and contacts. The servers themselves may be located at arbitrary locations on an internet. Thus, the tree of servers is an abstraction that uses an internet for communication.

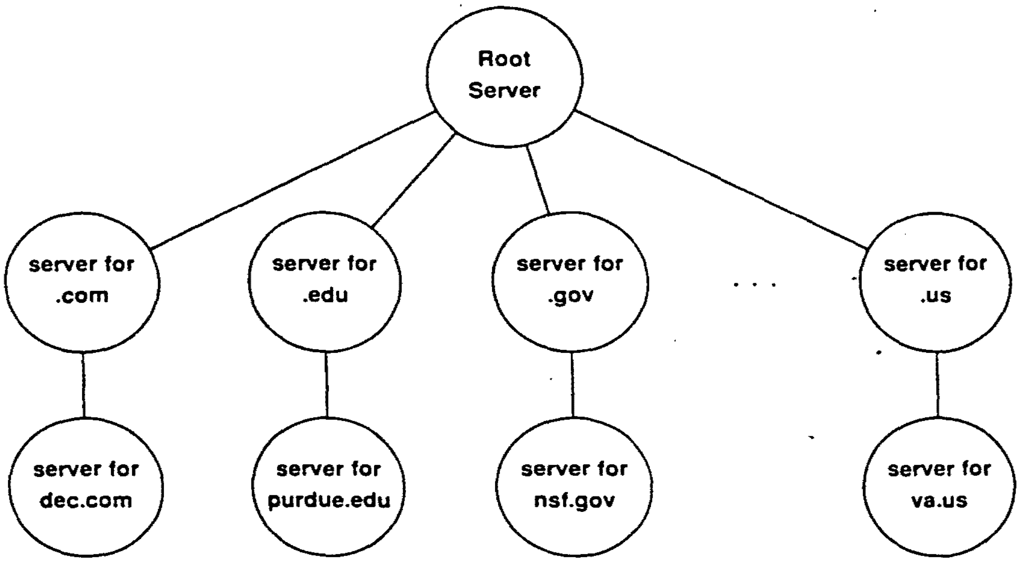


Figure 20.3 The conceptual arrangement of domain name servers in a tree that corresponds to the naming hierarchy. In theory, each server knows the addresses of all lower level servers for all subdomains within the domain it handles.

If servers in the domain system worked exactly as our simplistic model suggests, the relationship between connectivity and authorization would be quite simple. When authority was granted for a subdomain, the organization requesting it would need to establish a domain name server for that subdomain and link it into the tree.

In practice, the relationship between the naming hierarchy and the tree of servers is not as simple as our model implies. The tree of servers has few levels because a single physical server can contain all of the information for large parts of the naming hierarchy. In particular, organizations often collect information from all of their subdomains into a single server. Figure 20.4 shows a more realistic organization of servers for the naming hierarchy of Figure 20.2.

A root server contains information about the root and top-level domains, and each organization uses a single server for its names. Because the tree of servers is shallow, at most two servers need to be contacted to resolve a name like *xinu.cs.purdue.edu*: the root server and the server for domain *purdue.edu* (i.e., the root server knows which server handles *purdue.edu*, and the entire domain information for Purdue resides in its server).

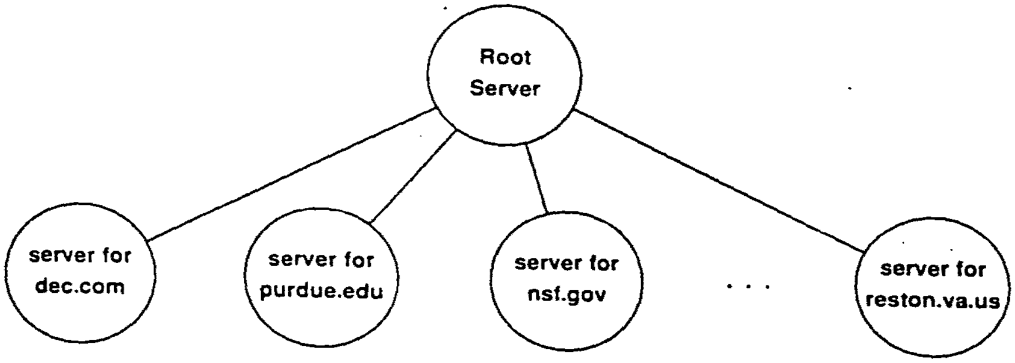


Figure 20.4 A realistic organization of servers for the naming hierarchy of Figure 20.2. Because the tree is broad and flat, few servers need to be contacted when resolving a name.

20.11 Domain Name Resolution

Although the conceptual tree makes understanding the relationship between servers easy, it hides several subtle details. Looking at the name resolution algorithm will help explain them. Conceptually, domain name resolution proceeds top-down, starting with the root name server and proceeding to servers located at the leaves of the tree. There are two ways to use the domain name system: by contacting name servers one at a time or asking the name server system to perform the complete translation. In either case, the client software forms a domain name query that contains the name to be resolved, a declaration of the class of the name, the type of answer desired, and a code that specifies whether the name server should translate the name completely. It sends the query to a name server for resolution.

When a domain name server receives a query, it checks to see if the name lies in the subdomain for which it is an authority. If so, it translates the name to an address according to its database and appends an answer to the query before sending it back to the client. If the name server cannot resolve the name completely, it checks to see what type of interaction the client specified. If the client requested complete translation (*recursive resolution*, in domain name terminology), the server contacts a domain name server that can resolve the name and returns the answer to the client. If the client requested non-recursive resolution (*iterative resolution*), the name server cannot supply an answer. It generates a reply that specifies the name server the client should contact next to resolve the name.

How does a resolver find a name server at which to begin the search? How does a name server find other name servers that can answer questions when it cannot? The answers are simple. A resolver must know how to contact at least one name server. To insure that a domain name server can reach others, the domain system requires that each server know the address of at least one root server†. In addition, a server may know the address of a server for the domain immediately above it (called the *parent*).

Domain name servers use a well-known protocol port for all communication, so clients know how to communicate with a server once they know the IP address of the machine in which the server executes. There is no standard way for hosts to locate a machine in the local environment on which a name server runs; that is left to whomever designs the resolver software‡.

In some systems, the address of the machine that supplies domain name service is bound into application programs at compile time, while in others, the address is configured into the operating system at startup. In others, the administrator places the address of a server in a file on secondary storage.

20.12 Efficient Translation

Although it may seem natural to resolve queries by working down the tree of name servers, it can lead to inefficiencies for three reasons. First, most name resolution refers to local names, those found within the same subdivision of the namespace as the machine from which the request originates. Tracing a path through the hierarchy to contact the local authority would be inefficient. Second, if each name resolution always started by contacting the topmost level of the hierarchy, the machine at that point would become overloaded. Third, failure of machines at the topmost levels of the hierarchy would prevent name resolution, even if the local authority could resolve the name. The telephone number hierarchy mentioned earlier helps explain. Although telephone numbers are assigned hierarchically, they are resolved in a bottom-up fashion. Because the majority of telephone calls are local, they can be resolved by the local exchange

†For reliability, there are multiple servers for each node in the domain server tree.

‡See BOOTP in Chapter 19 for one possible approach.

without searching the hierarchy. Furthermore, calls within a given area code can be resolved without contacting sites outside the area code. When applied to domain names, these ideas lead to a two-step name resolution mechanism that preserves the administrative hierarchy but permits efficient translation.

We have said that most queries to name servers refer to local names. In the two-step name resolution process, resolution begins with the local name server. If the local server cannot resolve a name, the query must then be sent to another server in the domain system.

20.13 Caching: The Key To Efficiency

The cost of lookup for nonlocal names can be extremely high if resolvers send each query to the root server. Even if queries could go directly to the server that has authority for the name, name lookup can present a heavy load to an internet. Thus, to improve the overall performance of a name server system, it is necessary to lower the cost of lookup for nonlocal names.

Internet name servers use *name caching* to optimize search costs. Each server maintains a cache of recently used names as well as a record of where the mapping information for that name was obtained. When a client asks the server to resolve a name, the server first checks to see if it has authority for the name according to the standard procedure. If not, the server checks its cache to see if the name has been resolved recently. Servers report cached information to clients, but mark it as a *nonauthoritative* binding, and give the domain name of the server, *S*, from which they obtained the binding. The local server also sends along additional information that tells the client the binding between *S* and an IP address. Therefore, clients receive answers quickly, but the information may be out-of-date. If efficiency is important, the client will choose to accept the nonauthoritative answer and proceed. If accuracy is important, the client will choose to contact the authority and verify that the binding between name and address is still valid.

Caching works well in the domain name system because name to address bindings change infrequently. However, they do change. If servers cached information the first time it was requested and never changed it, entries in the cache could become incorrect. To keep the cache correct, servers time each entry and dispose of entries that exceed a reasonable time. When the server is asked for the information after it has removed the entry from the cache, it must go back to the authoritative source and obtain the binding again. More important, servers do not apply a single fixed timeout to all entries, but allow the authority for an entry to configure its timeout. Whenever an authority responds to a request, it includes a *Time To Live* (TTL) value in the response that specifies how long it guarantees the binding to remain. Thus, authorities can reduce network overhead by specifying long timeouts for entries that they expect to remain unchanged, while improving correctness by specifying short timeouts for entries that they expect to change frequently.

Caching is important in hosts as well as local domain name servers. Many timesharing systems run a complex form of resolver code that attempts to provide even more efficiency than the server system. The host downloads the complete database of names and addresses from a local domain name server at startup, maintains its own cache of recently used names, and uses the server only when names are not found. Naturally, a host that maintains a copy of the local server database must check with the server periodically to obtain new mappings, and it must remove entries from its cache after they become invalid. However, most sites have little trouble maintaining consistency because domain names change so infrequently.

Keeping a copy of the local server's database in each host has several advantages. Obviously, it makes name resolution on local hosts extremely fast because it means the host can resolve names without any network activity. It also means that the local site has protection in case the local name server fails. Finally, it reduces the computational load on the name server, and makes it possible for a given server to supply names to more machines.

20.14 Domain Server Message Format

Looking at the details of messages exchanged between clients and domain name servers will help clarify how the system operates from the view of a typical application program. We assume that a user invokes an application program and supplies the name of a machine with which the application must communicate. Before it can use protocols like TCP or UDP to communicate with the specified machine, the application program must find the machine's IP address. It passes the domain name to a local resolver and requests an IP address. The local resolver checks its cache and returns the answer if one is present. If the local resolver does not have an answer, it formats a message and sends it to the server (i.e., it becomes a client). Although our example only involves one name, the message format allows a client to ask multiple questions in a single message. Each question consists of a domain name for which the client seeks an IP address, a specification of the query class (i.e., *internet*), and the type of object desired (e.g., *address*). The server responds by returning a similar message that contains answers to the questions for which the server has bindings. If the server cannot answer all questions, the response will contain information about other name servers that the client can contact to obtain the answers.

Responses also contain information about the servers that are authorities for the replies and the IP addresses of those servers. Figure 20.5 shows the message format.

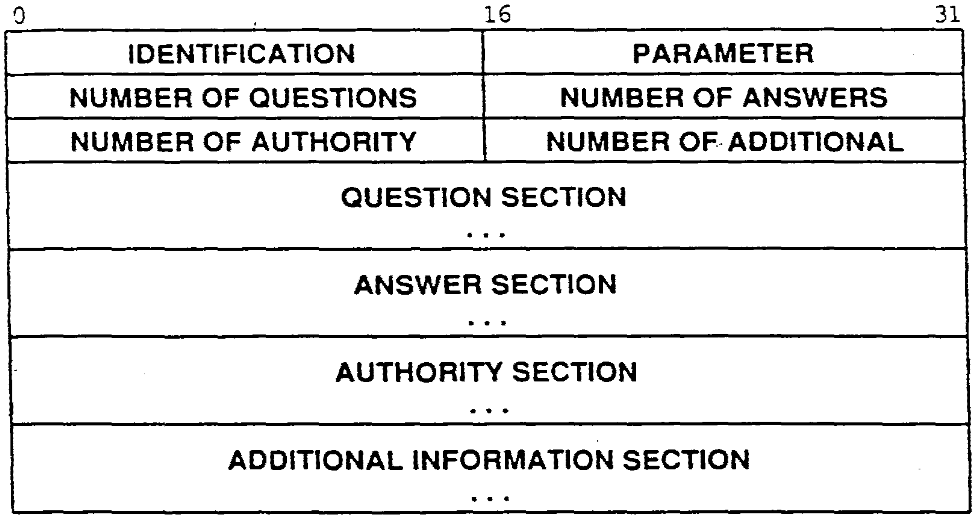


Figure 20.5 Domain name server message format. The question, answer, authority, and additional information sections are variable length.

As Figure 20.5 shows, each message begins with a fixed header that contains a unique *IDENTIFICATION* field that the client uses to match responses to queries. In the header, the field labeled *PARAMETER* specifies the operation requested and a response code, as shown in Figure 20.6 below.

The fields labeled *NUMBER OF* each give a count of entries in the corresponding sections that occur later in the message. For example, the field labeled *NUMBER OF QUESTIONS* gives the count of entries that appear in the *QUESTION SECTION* of the message.

The *QUESTION SECTION* contains queries for which answers are desired. The client fills in only the question section: the server returns the questions and answers in its response. Each question consists of a *QUERY DOMAIN NAME* followed by *QUERY TYPE* and *QUERY CLASS* fields, as Figure 20.7 shows.

Bit of PARAMETER field	Meaning
0	Operation:
	0 Query
	1 Response
1-4	Query Type:
	0 Standard
	1 Inverse
	2 Completion 1 (now obsolete)
	3 Completion 2 (now obsolete)
5	Set if answer authoritative
6	Set if message truncated
7	Set if recursion desired
8	Set if recursion available
9-11	Reserved
12-15	Response Type:
	0 No error
	1 Format error in query
	2 Server failure
	3 Name does not exist

Figure 20.6 The meaning of bits of the *PARAMETER* field in a domain name server message. Bits are numbered left to right starting at 0.

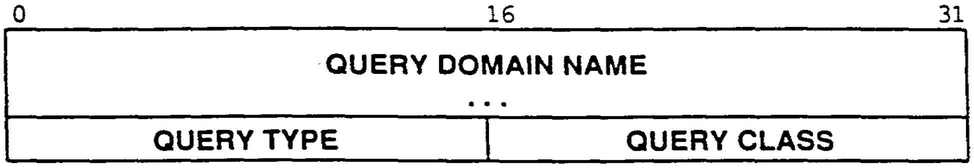


Figure 20.7 The format of entries in the question section of a domain name server message. The domain name is variable length. Clients fill in the questions; servers return them along with answers.

Although the *QUERY DOMAIN NAME* field has variable length, we will see in the next section that the internal representation of domain names makes it possible for the receiver to know the exact length. The *QUERY TYPE* encodes the type of the question (e.g., whether the question refers to a machine name or a mail address). The *QUERY CLASS* field allows domain names to be used for arbitrary objects because official Internet names are only one possible class. It should be noted that, although the diagram in Figure 20.5 follows our convention of showing formats in 32-bit multiples, the query domain name field may contain an arbitrary number of octets. No padding is used. Therefore, messages to or from domain name servers may contain an odd number of octets.

In a domain name server message, each of the *ANSWER SECTION*, *AUTHORITY SECTION*, and *ADDITIONAL INFORMATION SECTION* consists of a set of *resource records* that describe domain names and mappings. Each resource record describes one name. Figure 20.8 shows the format.

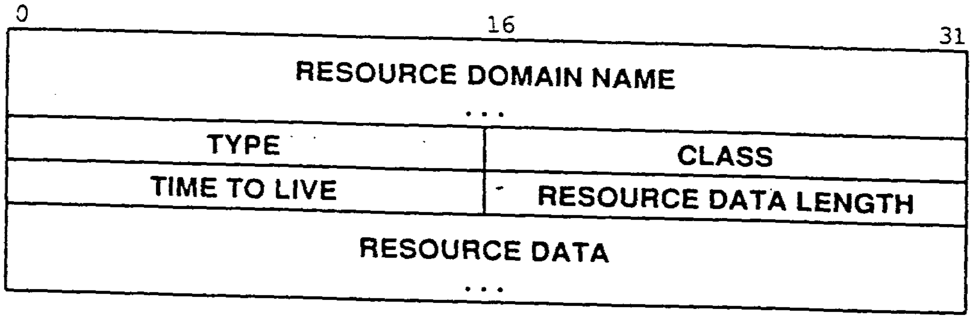


Figure 20.8 The format of resource records used in later sections of messages returned by domain name servers.

The *RESOURCE DOMAIN NAME* field contains the domain name to which this resource record refers. It may be an arbitrary length. The *TYPE* field specifies the type of the data included in the resource record; the *CLASS* field specifies it class. The *TIME TO LIVE* field contains an integer that specifies the number of seconds information in this resource record can be cached. It is used by clients who have requested a name binding and may want to cache the results. The last two fields contain the results of the binding, with the *RESOURCE DATA LENGTH* field specifying the count of octets in the *RESOURCE DATA* field.

20.15 Compressed Name Format

When represented in a message, domain names are stored as a sequence of labels. Each label begins with an octet that specifies its length. Thus, the receiver reconstructs a domain name by repeatedly reading a 1-octet length, *n*, and then reading a label *n* octets long. A length octet containing zero marks the end of the name.

Domain name servers often return multiple answers to a query and, in many cases, suffixes of the domain overlap. To conserve space in the reply packet, the name servers compress names by storing only one copy of each domain name. When extracting a domain name from a message, the client software must check each segment of the name to see whether it consists of a literal string (in the format of a 1-octet count followed by the characters that make up the name) or a pointer to a literal string. When it encounters a pointer, the client must follow the pointer to a new place in the message to find the remainder of the name.

Pointers always occur at the beginning of segments and are encoded in the count byte. If the top 2 bits of the 8-bit segment count field are 1s, the client must take the next 14 bits as an integer pointer. If the top two bits are zero, the next 6 bits specify the number of characters in the label that follow the count octet.

20.16 Abbreviation Of Domain Names

The telephone number hierarchy illustrates another useful feature of local resolution, viz., *name abbreviation*. Abbreviation provides a method of shortening names when the resolving process can supply part of the name automatically. Normally, a subscriber omits the area code when dialing a local telephone number. The resulting digits form an abbreviated name assumed to lie within the same area code as the subscriber's phone. Abbreviation also works well for machine names. Given a name like *xyz*, the resolving process can assume it lies in the same local authority as the machine on which it is being resolved. Thus, the resolver can supply missing parts of the name automatically. For example, within the Computer Science Department at Purdue, the abbreviated name

xinu

is equivalent to the full domain name

xinu . cs . purdue . edu

Most client software implements abbreviations with a *domain suffix list*. The local network manager configures a list of possible suffixes to be appended to names during lookup. When a resolver encounters a name, it steps through the list, appending each suffix and trying to look up the resulting name. For example, the suffix list for the Computer Science Department at Purdue includes:

.cs.purdue.edu
.cc.purdue.edu
.purdue.edu
null

Thus, local resolvers first append *cs.purdue.edu* onto the name *xinu*. If that lookup fails, they append *cc.purdue.edu* onto the name and look that up. The last suffix in the example list is the null string, meaning that if all other lookups fail, the resolver will attempt to look up the name with no suffix. Managers can use the suffix list to make abbreviation convenient or to restrict application programs to local names.

We said that the client takes responsibility for the expansion of such abbreviations. but it should be emphasized that such abbreviations are not part of the domain name system itself. The domain system only allows lookup of a fully specified domain name. As a consequence. programs that depend on abbreviations may not work correctly outside the environment in which they were built. We can summarize:

> *The domain name system only maps full domain names into addresses: abbreviations are not part of the domain name system itself, but are introduced by client software to make local names convenient for users.*

20.17 Inverse Mappings

We said that the domain name system can provide mappings other than machine name to IP address. *Inverse queries* allow the client to ask a server to map "backwards" by taking an answer and generating the question that would produce that answer. Of course. not all answers have a unique question. Even when they do. a server may not be able to provide it. Although inverse queries have been part of the domain system since it was first specified. they are generally not used because there is often no way to find the server that can resolve the query without searching the entire set of servers.

20.18 Pointer Queries

One form of inverse mapping is so obviously needed that the domain system supports a special domain and a special form of question called a *pointer query* to answer it. In a pointer query. the question presented to a domain name server specifies an IP address encoded as a printable string in the form of a domain name (i.e., a textual representation of digits separated by periods). A pointer query requests the name server to return the correct domain name for the machine with the specified IP address. Pointer queries are especially useful for diskless machines because they allow the system to obtain a high-level name given only an IP address. (We have already seen in Chapter 6 how a diskless machine can obtain its IP address.)

Pointer queries are not difficult to generate. If we think of an IP address written in dotted-decimal form. it has the following format:

aaa.bbb.ccc.ddd

To form a pointer query. the client rearranges the dotted decimal representation of the address into a string of the form:

ddd.ccc.bbb.aaa.in-addr.arpa

The new form is a name in the special *in-addr.arpa* domain[†]. Because the local name server may not be the authority for either the *arpa* domain or the *in-addr.arpa* domain, it may need to contact other name servers to complete the resolution. To make the resolution of pointer queries efficient, the Internet root domain servers maintain a database of valid IP addresses along with information about domain name servers that can resolve each address.

20.19 Object Types And Resource Record Contents

We have mentioned that the domain name system can be used for translating a domain name to a mail exchanger address as well as for translating a host name to an IP address. The domain system is quite general in that it can be used for arbitrary hierarchical names. For example, one might decide to store the names of available computational services along with a mapping from each name to the telephone number to call to find out about the corresponding service. Or one might store names of protocol products along with a mapping to the names and addresses of vendors that offer such products.

Recall that the system accommodates a variety of mappings by including a *type* in each resource record. When sending a request, a client must specify the type in its query[†]; servers specify the data type in all resource records they return. The type determines the contents of the resource record according to the table in Figure 20.9

Type	Meaning	Contents
A	Host Address	32-bit IP address
CNAME	Canonical Name	Canonical Domain Name for an alias
HINFO	CPU & OS	Name of CPU and Operating System
MINFO	Mailbox info	Information about a mailbox or mail list
MX	Mail Exchanger	16-bit preference and name of host that acts as mail exchanger for the domain
NS	Name Server	Name of authoritative server for domain
PTR	Pointer	Domain name (like a symbolic link)
SOA	Start of Authority	Multiple fields that specify which parts of the naming hierarchy a server implements
TXT	Arbitrary text	Uninterpreted string of ASCII text

Figure 20.9 Domain Name System resource record types.

Most data is of type *A*, meaning that it consists of the name of a host attached to the Internet along with the host's IP address. The second most useful domain type, *MX*, i: assigned to names used for electronic mail exchangers. It allows a site to specify multiple machines that are each capable of accepting mail. When sending electronic mail the user specifies an electronic mail address in the form *user@domain-part*. The ma

[†] The octets of the IP address must be reversed when forming a domain name because IP addresses ha the most significant octets first while domain names have the least-significant octets first.

[†] Queries can specify a few additional types (e.g., there is a query type that requests all resource records

system uses the domain name system to resolve *domain-part* with query type *MX*. The domain system returns a set of resource records that each contain a preference field and a host's domain name. The mail system steps through the set from highest preference to lowest (lower numbers mean higher preference). For each *MX* resource record, the mailer extracts the domain name and uses a type *A* query to resolve that name to an IP address. It tries to contact the host and deliver mail. If the host is unavailable, the mailer will continue trying other hosts on the list.

To make lookup efficient, a server always returns additional bindings that it knows in the *ADDITIONAL INFORMATION SECTION* of a response. In the case of *MX* records, a domain server can use the *ADDITIONAL INFORMATION SECTION* to return type *A* resource records for domain names reported in the *ANSWER SECTION*. Doing so substantially reduces the number of queries a mailer sends to its domain server.

20.20 Obtaining Authority For A Subdomain

Before an institution is granted authority for an official second-level domain, it must agree to operate a domain name server that meets Internet standards. Of course, a domain name server must obey the protocol standards that specify message formats and the rules for responding to requests. The server must also know the addresses of servers that handle each subdomain (if any exist) as well as the address of at least one root server.

In practice, the domain system is much more complex than we have outlined. In most cases, a single physical server may handle more than one part of the naming hierarchy. For example, a single name server at Purdue University handles both the second-level domain *purdue.edu* as well as the geographic domain *laf.in.us*. A subtree of names managed by a given name server forms a *zone of authority*. Another practical complication arises because servers must be able to handle many requests, even though some requests take a long time to resolve. Usually, servers support concurrent activity, allowing work to proceed on later requests while earlier ones are being processed. Handling requests concurrently is especially important when the server receives a recursive request that forces it to send the request on to another server for resolution.

Server implementation is also complicated because the Internet authority requires that the information in every domain name server be replicated. Information must appear in at least two servers that do not operate on the same computer. In practice, the requirements are quite stringent: the servers must have no single common point of failure. Avoiding common points of failure means that the two name servers cannot both attach to the same network: they cannot even obtain electrical power from the same source. Thus, to meet the requirements, a site must find at least one other site that agrees to operate a backup name server. Of course, at any point in the tree of servers, a server must know how to locate both the primary and backup name servers for subdomains, and it must direct queries to a backup name server if the primary server is unavailable.

20.21 Summary

Hierarchical naming systems allow delegation of authority for names, making it possible to accommodate an arbitrarily large set of names without overwhelming a central site with administrative duties. Although name resolution is separate from delegation of authority, it is possible to create hierarchical naming systems in which resolution is an efficient process that starts at the local server even though delegation of authority always flows from the top of the hierarchy downward.

We examined the Internet domain name system, an example of a distributed, hierarchical naming scheme. Domain name servers map high-level domain names to IP addresses or to mail exchanger addresses. Clients begin by trying to resolve names locally. When the local server cannot resolve the name, the client must choose to work through the tree of name servers iteratively or request the local name server to do it recursively. Finally, we saw that the domain name system supports a variety of bindings including bindings from IP addresses to high-level names.

FOR FURTHER STUDY

Mockapetris [RFC 1034] discusses Internet domain naming in general, giving the overall philosophy, while Mockapetris [RFC 1035] provides a protocol standard for domain naming. Mockapetris [RFC 1101] discusses using the domain name system to encode network names and proposes extensions useful for other mappings. Older versions appeared in Mockapetris [RFC 882; 883, and 973]. Postel and Reynolds [RFC 920] states the requirements that an Internet domain name server must meet. Stahl [RFC 1032] gives administrators guidelines for establishing a domain, and Lottor [RFC 1033] provides guidelines for operating a domain name server. Finally, Partridge [RFC 974] relates domain naming to electronic mail addressing.

EXERCISES

20.1 Machine names should not be bound into the operating system at compile time. Explain why.

20.2 Would you prefer to use a machine that obtained its name from a remote file or from a name server? Why?

20.3 Why should each name server know the IP address of its parent instead of the domain name of its parent?

20.4 Devise a naming scheme that tolerates changes to the naming hierarchy. As an example, consider that two large companies each have an independent hierarchy and they merge. Can you arrange to have all previous names still work correctly?

20.5 Read the standard and find out how the domain name system uses *SOA* records.

20.6 The Internet domain naming system can also accommodate mailbox names. Find out how.

20.7 The standard suggests that when a program needs to find the domain name associated with an IP address, it should send an inverse query to the local server first and use domain *in-addr.arpa* only if that fails. Why?

20.8 How would you accommodate abbreviations in the domain naming scheme? Sketch name servers for two departments at each of two universities as well as a top-level name server. Explain how each server would treat each type of abbreviation.

20.9 Obtain the official description of the domain name system and build a client program. Look up the name *merlin.cs.purdue.edu*.

20.10 Extend the exercise above to include a pointer query. Try looking up the domain name for address *128.10.2.3*.

20.11 If we extended the domain name syntax to include a dot after the top-level domain, names and abbreviations would be unambiguous. What are the advantages and disadvantages of the extension?

#13

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 556 012 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **93300919.3**

(22) Date of filing : **09.02.93**

(51) Int. Cl.^{5} : **H04Q 7/00,** H04L 12/56

(30) Priority : **10.02.92 JP 23506/92**
16.09.92 JP 246855/92
10.11.92 JP 299531/92

(43) Date of publication of application :
18.08.93 Bulletin 93/33

(84) Designated Contracting States :
DE FR GB

(71) Applicant : **MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.**
1006, Kadoma
Kadoma-shi, Osaka-fu 571 (JP)

(72) Inventor : **Wada, Hiromi**
15-10, Higashigaoka, Uzumasa
Neyagawa-shi, Osaka 572 (JP)
Inventor : **Yozawa, Takashi**
5-16-19, Shinke, Aoo
Mino-shi, Osaka 562 (JP)
Inventor : **Ohnishi, Tatsuya**
281-5, Kawahara, Aza, Sasabe
Kawanishi-shi, Hyogo 666-01 (JP)

(74) Representative : **Cummings, Sean Patrick et al**
David Keltie Associates Audrey House Ely Place
London EC1N 6SN (GB)

(54) Migration communication control device.

(57) Disclosed is a migration communication control device constructed to control a continuous communication between a mobile node and a node unaffected the mobile node's migration. The migration communication control device comprises a first migration control unit, a second migration control unit on the mobile node, and a third migration control unit on the partner node. The first migration control unit comprises a packet transfer unit and an address post unit. The packet transfer unit receives a packet which was destined for an outdated address of the mobile node, generates a conversion packet which holds an updated address instead of the outdated address, and then transmits the conversion packet, while an address post unit transmits an address post message which indicates the updated address to the third migration control unit. The second migration control unit comprises a migration post unit and a packet resumption unit. The migration post unit transmits to the first migration control unit a migration post message which indicates the updated address when the mobile node migrates to another network while a packet resumption unit receives the conversion packet from both the first migration control unit and the third migration control unit and resumes an original packet from the conversion packet. The third migration control unit comprises a packet conversion unit which converts a destination address of a packet into the updated address, then transmits it to the mobile node.

EP 0 556 012 A2

BACKGROUND OF THE INVENTION

(1) Field of the Invention

The present invention relates to a migration communication control device that controls a communication between a mobile node and a corresponding node to enable them to communicate continuously when the former migrates by managing addresses assigned to the former each time it migrates across networks.

(2) Description of the Related Art

Recent progress in the field of electronic technology makes it possible to assemble smaller and lighter portable computers. These portable computers referred to as mobile nodes are designed so that they can migrate across networks: they are unplugged from a network and plugged in another and communicate with a stationary node. Thus, each of them is assigned a specific address to prove its identity. The address, in general, includes location information as to which network the mobile nodes are currently plugged in, and for this reason, a new address is assigned each time they migrate.

For example, the address composed of a network address unit for specifying a network in which the mobile node is currently plugged in and a node address unit for proving the mobile node's identity in the network, or the address used in a conventional network architecture such as Internet Protocol(details of which are in Internet Protocol, RFC791, Jon Postel, Sep., 1981), they must be changed every time the mobile nodes migrate.

However, once the mobile node migrates to another network, a communication with the stationary node will be terminated. This is because a packet is transmitted to its old address only to be wasted.

Thus, to enable the mobile node and stationary node to communicate continuously when the former migrates, it is necessary to control the communication by managing the steadily changing address.

To date, two address managing methods have been proposed: one by Sony Computer Science Laboratory Inc. and one by the Department of Computer Science at Columbia University.

Sony Computer Science Laboratory Inc. proposed a method using VIP(Virtual Internet Protocol), details of which are on "VIP : Lower Layer Internet Protocol", Fumio Teraoka, Yasuhiko Yokote, Mario Tokoro, Proceed of Data Processing Convention : Multimedia Communication and Distributed Processing.

In this method, each mobile node is assigned a VIP(Virtual Internet Protocol) address and a PIP(Physical Internet Protocol) address. The former is an unchanged address used in a communication application for packet transmission and reception; and the latter is an address changed for every migration to specify an update physical location of the mobil node. Data related to both addresses are held in a cache of a gateway. Under these conditions, the stationary node transmits a packet to the mobile node to the VIP address thereof, and the packet is converted into another packet addressed to the PIP address when it passes the gateway, thence transmitted to the mobile node via the gateways placed in a route onwards. These gateways collect data related to a correlation between the VIP and PIP addresses from the header of the packet upon the receipt thereof, thus updates data in the cache, and hence are able to convert other packets addressed to the VIP addresses into the packets addressed to the PIP addresses based on the correlation entered in the cache.

In this method, in short, the use of the address constituting with the VIP and PIP addresses enables the mobile node and the stationary node to communicate continuously when the former migrates.

The Department of Computer Science at Columbia University proposed a method using an Internet Protocol address of which network address unit does not specify the network which the mobile node is currently plugged in but declares itself to be the mobile node, hence a certain value is given as the network address unit to all the mobile nodes. As well, the method uses an MSS(Mobile Support Station) installed at each network to manage the IP addresses and control a packet route to the mobile node. The MSS is designed so that it collects data related to the update physical location of the mobile nodes by referring other MSSs.

Given these conditions, when the stationary node transmits a packet to the mobile node when it migrates, it first transmits the packet to a first MSS installed in its network; thence the first MSS transfers the packet to a second MSS installed in a network which the mobile node is currently plugged in; and finally the second MSS transfers the packet to the mobile node.

In this method, in short, the use of the MSS enables the mobile node and the stationary node to continue the communication when the former migrates.

In the first method, however, all the nodes must be constructed so that they understand both the VIP and PIP addresses, causing them to extend a scale functionally, otherwise making it impossible to apply this method to apparatuses employed in existing networks. In addition, the communication via the gateways reduces communication efficiency compared with direct packet transmission, because the gateways check whether they have received the packet addressed to the VIP address or PIP address each time they receive it, as well as whether or not to collect the data therefrom to update those in the cache.

In the second method, each network must have the MSS, and the communication via the MSSs makes it impossible to transmit the packet directly, thereby reducing the communication efficiency.

SUMMARY OF THE INVENTION

The present invention therefore has an object to provide a migration communication control device that is available to any apparatus employed in existing networks. Also the present invention has another object to provide a migration communication control device that enables the mobile node and stationary node to communicate continuously when the former migrates by transmitting and receiving the packet directly besides transferring the packet as has been done when the mobile node migrates across the networks.

The above objects are fulfilled by a migration communication control device constructed to control a communication between a mobile node and a partner node, the mobile node migrating across networks and obtaining an address assigned on each network while the partner node being a communication partner of the mobile node, comprising a first migration control unit, a second migration control unit, a third migration control unit, the second migration control unit being placed on the mobile node and the third migration control unit being placed on the partner node, wherein the first migration control unit comprises a packet transfer unit for receiving a packet which was destined for an outdated address of the mobile node, the outdated address assigned when the mobile node migrated to a network to which the first migration control unit is attached, generating a conversion packet which holds an updated address instead of the outdated address, and transmitting the conversion packet; and an address post unit for transmitting an address post message which indicates the updated address of the mobile node to the third migration control unit, the third migration control unit transmitting the packet received by the packet transfer unit, and the second migration control unit comprises a migration post unit for transmitting to the first migration control unit a migration post message which indicates the updated address of the mobile node when the mobile node migrates to another network; and a packet resumption unit for receiving the conversion packet from both the first migration control unit and the third migration control unit and resuming an original packet from the conversion packet, and the third migration control unit comprises a packet conversion unit for converting a destination address of a packet, the packet to be transmitted to the mobile node, into the updated address indicated by the address post message, the address post message sent by the first migration control unit, and transmitting it to the mobile node.

The migration post unit in the second migration control unit may transmit an identification key included in the migration post message, the identification key being employed to identify the mobile node.

The identification key may be an address of the mobile node assigned at one network before the network to which the mobile node is currently attached.

The identification key may be an address of the mobile node assigned before its initial migration.

The second migration control unit may be constructed to transmit to the third migration control unit the packet which has the same format as the resumed packet.

The first migration control unit may further comprise an address hold unit for holding the outdated address and the updated address by corresponding them with each other; and an address comparison unit for comparing the destination address of the received packet with the outdated address, wherein the packet transfer unit generates the conversion packet and transmits it when the address comparison unit detects that the destination address of the received packet coincides with the outdated address.

The first migration control unit may further comprise an address hold unit for holding the outdated address and the updated address by corresponding them with each other; and an address comparison unit for comparing the destination address of the packet received by the packet transfer unit with the outdated address, wherein the address post unit transmits the address post message which indicates the updated address of the mobile node to the third migration control unit, the third migration control unit transmitting the packet received by the packet transfer unit, when the address comparison unit detects that the destination address of the packet coincides with the outdated address.

The second migration control unit may further comprise an address hold unit for holding the outdated address and the updated address by corresponding them with each other; and an address comparison unit for comparing the updated address with the destination address of the packet received from one of the first migration control unit and the third migration control unit, wherein the packet resumption unit resumes the original packet from the conversion packet when the address comparison unit detects that the updated address coincides with the destination address of the packet received from one of the first migration control unit and the third migration control unit.

The third migration control unit may further comprise an address hold unit for holding the outdated address and the updated address of the mobile node by corresponding them with each other; and an address comparison unit for comparing the outdated address in the address hold unit with the destination address of the packet to be transmitted to the mobile node, wherein the packet conversion unit converts the destination address of the packet to be transmitted to the mobile node into the updated address which corresponds to the outdated address in the address hold unit when the address comparison unit detects the outdated address in the address hold unit coincides with the destination address of the packet.

There may be a plurality of the first migration control units, and the second migration control unit transmits the migration post message to at least one of the first migration control units.

The migration post unit in the second migration control unit may transmit the migration post message to the first migration control unit which is attached to the network to which the mobile node was attached before its migration, each of the first migration control units has a migration post unit for transmitting to one of the other first migration control units a migration post message to post the same address as the updated address indicated by the migration post message received from the second migration control unit, and each of the first migration control units has a migration post unit for transmitting a migration post message from one of the other first migration control units to another first migration control unit to post the same address as the updated address indicated by the received migration post message.

Each of the first migration control units and the second migration control unit may further comprise a pointer hold unit for holding pointers related to the first migration control unit to which the migration post message is transmitted, and wherein the migration post unit in each of the first migration control units and the migration post unit in the second migration control unit transmit the migration post message to each of the addresses related to each of the pointers.

Each of the pointers may be a broadcast address of the network to which one of the first migration control units is attached.

Each of the pointers may be an address which is assigned to one of the first migration control units uniquely.

Each of the pointers may be the address of the mobile node which is assigned when the mobile node is attached to the same network as is the first migration control unit, and the migration post unit in the first migration control unit and the migration post unit in the second migration control unit obtain the broadcast address of the network to which each of the first migration control units is attached with referring to the address of the mobile node, and transmits the migration post message to the obtained broadcast address.

The pointer hold unit in the second migration control unit may hold a pointer related to a first migration control unit for the latest migration, which is the first migration control unit being attached to one network before the network to which the mobile node is currently attached, and the pointer hold unit in the first migration control unit holds a pointer related to another first migration control unit attached to the same network as was the mobile node attached before migrating to the network to which the first migration control unit is attached.

The second migration control unit may further transmit to the first migration control unit the pointer by sending thereto the migration post message, the pointer to be held by the first migration control unit.

The first migration control unit may store into the pointer hold unit the pointer when it receives from the second migration control unit the migration post message by corresponding the pointer with the updated address indicated by the received migration post message.

Each of the first migration control units may further comprise an address hold unit for holding the outdated address and the updated address by corresponding them with each other, wherein a migration post message unit stores into the address hold unit the outdated address and the updated address by corresponding them with each other when it receives from the second migration control unit the migration post message, while converts the updated address in the address hold unit into the updated address indicated by the migration post message when it receives from the first migration control unit the migration post message and the outdated address indicated by the migration post message coincides with one of the updated addresses in the address hold unit.

The first migration control unit may be placed on a gateway, which connects networks.

The first migration control unit may be placed on the network as an individual node.

The migration post unit in the second migration control unit may transmit the migration post message to a home migration control unit, the home migration control unit being the first migration control unit which is attached to a network where the mobile node left for its initial migration, and the home migration control unit may further comprise a home migration post unit for transmitting a migration post message to a first migration control unit for the latest migration, the first migration control unit for the latest migration being the first migration control unit which is attached to the network where the mobile node left for the latest migration, to post the same updated address as is indicated by the migration post message received from the second migration control unit.

The first migration control unit may further comprise a migration post unit for transmitting the migration post message indicating the updated address of the mobile node to one of the other first migration control units when the conversion.packet destined for the outdated address of the mobile node was sent therefrom to the first migration control unit.

The migration post unit in the second migration control unit may transmit to the home migration control unit the migration post message where a home address and the updated address are corresponded with each other, the home address assigned when the mobile node is attached to the same network as is the home migration control unit, and each of the packet transfer unit and the address post unit in the home migration control unit may transmit the conversion packet and the address post message respectively with referring to the above home address and the updated address.

The second migration control unit may further comprise an outdated address post unit for transmitting to the first migration control unit for the latest migration an outdated address post message where the outdated address and the home address are corresponded with each other, the outdated address being assigned to the mobile node before the latest migration, the home migration post unit in the home migration control unit may transmit to the said first migration control unit for the latest migration the migration post message where the above home address and the updated address are corresponded with each other, and the packet transfer unit and the address post unit in the first migration control unit for the latest migration may transmit the conversion packet and the address post message respectively in accordance with the outdated address and the updated address, the outdated address and the updated address being corresponded with each other via the home address.

The outdated address post unit in the second migration control unit may transmit the above outdated address post message at a migration of the mobile node preceding the latest migration, and each of the migration post units in the second migration control unit and the home migration post unit in the home migration control unit may transmit the above migration post message at the latest migration of the mobile node.

The second migration control unit may further comprise a home migration control unit pointer hold unit for holding a pointer related to the home migration control unit, the migration post unit in the second migration control unit transmits the migration post message to the address related to the pointer, the home migration control unit may further comprise a pointer hold unit for the latest migration for holding a pointer related to the first migration control unit for the latest migration, and the home migration post unit in the home migration control unit may transmit the migration post message to the address related to the pointer.

Each of the above pointers may be the broadcast address of the network to which each of the first migration control units is attached.

Each of the above pointers may be the address assigned to each of the first migration control units uniquely.

The second migration control unit may further comprise a pointer obtainment unit for requesting to the first migration control unit for the latest migration the pointer related to the first migration control unit for the latest migration, and the migration post unit in the second migration control unit may post the obtained pointer to the home migration control unit together with the updated address by sending thereto the migration post message.

The migration post unit in the second migration control unit may post to the home migration control unit the pointer at the migration of the mobile node preceding the latest migration, while the migration post unit may post the above updated address at the latest migration of the mobile node.

The first migration control unit may further comprise an address post suppressing unit for suppressing transmission of the address post message from the address post unit to the third migration control unit, and the address post suppressing unit may suppress transmission of the address post message when none of the first migration control units is attached to the same network as is the mobile node.

The second migration control unit may further comprise a detect unit for detecting whether or not the first migration control unit is attached to the network to which the mobile node migrates, the migration post unit in the second migration control unit may transmit to the home migration control unit the migration post message which includes the detecting result of the above detect unit together with the updated address, the home migration post unit in the home migration control unit may transmit to the first migration control unit for the latest migration the migration post message which includes the detecting result of the above detect unit together with the updated address, and the address post suppres^sing unit in each of the home migration control unit and the first migration control unit for the latest migration may suppress the transmission of the address post message in accordance with the detecting result of the above detect unit.

The first migration control unit may further comprise a packet transfer suppressing unit for suppressing transfer of the packet conducted by the packet transfer unit.

The first migration control unit may further comprise an address post suppressing unit for suppressing transmission of the address post message from the address post unit to the third migration control unit, and the address post suppressing unit in the first migration control unit being attached to a network to which the mobile node is not attached, may suppress the transmission of the address post message when the packet transfer suppressing unit in the first migration control unit for the latest migration suppresses transfer of the packet.

The second migration control unit may further comprise a detect unit for detecting whether or not the packet transfer suppressing unit in the first migration control unit suppresses the transfer of the packet, the first migration control unit being attached to the network to which the mobile node migrates, and the migration post unit in the second migration control unit transmits to the home migration control unit the migration post message which includes the detecting result of the above detect unit together with the updated address, the home migration post unit in the home migration control unit may transmit to the first migration control unit for the latest migration the migration post message which includes the detecting result of the detect unit together with the updated address, and the address post suppressing unit in each of the home migration control unit and the first migration control unit for the latest migration may suppress the transmission of the address post message in accordance with the detecting result of the above detect unit.

The packet transfer suppressing unit in the first migration control unit for the latest migration may suppress the transfer of the packet conducted by the packet transfer unit, when the packet transfer suppressing unit in the first migration control unit being attached to the network to which the mobile node migrates suppresses the transfer of the packet.

The above objects may also be fulfilled by a packet transfer migration control unit in a migration communication control device, the migration communication control device being constructed to control a communication between a mobile node and a partner node, the mobile node migrating across networks and obtaining an address assigned on each network while the partner node being a communication partner of the mobile node, comprising a packet transfer unit for receiving a packet which was transmitted by the partner node to an outdated address of the mobile node, the outdated address being assigned when the mobile node migrated to a network to which the packet transfer migration control unit is attached, generating a conversion packet which holds an updated address instead of the outdated address, and transmitting the conversion packet; and an address post unit for transmitting an address post message which indicates the updated address of the mobile node to the partner node, the partner node transmitting the packet received by the packet transfer unit.

The above objects may further be fulfilled by a mobile node migration control unit in a migration communication control device, the migration communication control device being constructed to control a communication between a mobile node which migrates across networks and obtains an address assigned on each network and a partner node which is a communication partner of the mobile node, being placed on the mobile node and comprising a migration post unit for transmitting to a packet transfer migration control unit a migration post message which indicates an updated address of the mobile node when the mobile node migrates to another network, the packet transfer migration control unit for receiving a packet which was transmitted by the partner node to an outdated address of the mobile node, the outdated address assigned when the mobile node migrated to a network to which the migration control unit for packet transfer is attached, generating a conversion packet which holds the updated address instead of the outdated address, and transmitting the conversion packet; and a packet resumption unit for receiving the conversion packet from both the packet transfer migration control unit and the mobile node, and resuming an original packet from the conversion packet.

The above objects are finally fulfilled by a partner node migration control unit in a migration communication control device, the migration communication control device being constructed to control a communication between a mobile node which migrates across networks and obtains an address assigned on each network and a partner node which is a communication partner of the mobile node, being placed on the mobile node and comprising an address post message receiving unit for receiving an address post message which indicates an updated address of the mobile node from a packet transfer migration control unit, the packet transfer migration control unit transmitting an address post message which indicates the updated address of the mobile node to the partner node; and a packet conversion unit for converting a destination address of a packet, the packet to be transmitted to the mobile node, into the updated address indicated by the address post message, and transmitting it to the mobile node.

According to the above construction, the migration communication control device of the present invention transfers and converts the packet using the address assigned to the mobile node each time it migrates across networks, obviating particular addresses or devices such as the VIP address used conventionally. For this reason, the migration communication control device of the present invention can be applied to the existing partner node and mobile node so that they can communicate continuously by transferring the packet. Moreover, it is advantageous that the migration communication control device of the present invention is not necessarily applied to all the nodes to enhance communication efficiency; the present invention can be applied only to where necessary on the existing networks. More precisely, when any existing partner node communicates with the mobile node when it migrates, the packet can be transmitted directly from the mobile nodes to the existing partner node; and it can be transferred via the first migration control unit from the existing partner node to the mobile node, thereby enhancing communication efficiency.

Furthermore, when the partner node employs the migration communication control device of the present invention, communication efficiency is further enhanced thanks to the direct packet transmission and reception made possible by posting the update address of the mobile node from the first migration control unit to the third migration control unit.

Also, the devices such as MSS or a gateway employing the VIP are not necessarily installed at every network to which the mobile node migrates. To be precise, according to the present invention, the continuous communication is implemented even when the mobile node migrates to a network at which no special devices including above ones are installed.

BRIEF DESCRIPTION OF THE DRAWINGS

These and the other objects, advantages and features of the invention will become apparent from the following description thereof taken in conjunction with the accompanying drawings which illustrate a specific embodiment of the invention. In the drawings:

FIG. 1 is a block diagram depicting a construction of a migration communication control device in a first embodiment of the present invention;

FIG. 2 is a block diagram depicting a detailed construction of the migration communication control device employed as a mobile node in the first embodiment of the present invention;

FIG. 3 is a block diagram depicting a detailed construction of the migration communication control device employed as a gateway in the first embodiment of the present invention;

FIG. 4 is a block diagram showing a detailed construction of the migration communication control device employed as a stationary node in the first embodiment of the present invention;

FIG. 5 is a block diagram showing a detailed construction of the migration communication control device employed as an individual node in the first embodiment of the present invention;

FIG. 6 is an illustration showing a first example of a network to which the migration communication control devices in FIG. 2, 3, 4 are attached;

FIG. 7 is an illustration showing a second example of the network to which the migration communication control devices in FIG. 2, 3, 4 are attached;

FIG. 8 is an illustration showing a third example of the network to which the migration communication control devices in FIG. 2, 3, 4 are attached;

FIG. 9 is an illustration showing a fourth example of the network to which the migration communication control devices in FIG. 2, 3, 4 are attached;

FIG. 10 is an illustration showing (a) data in a data hold unit 1 in the mobile node (b) data in a data hold unit 1 in the migration communication control devices each employed as the gateway, the stationary node, and the individual node.

FIG. 11 is an illustration showing a format of a packet in the first embodiment of the present invention;

FIG. 12 is an illustration showing a format of a packet in the first embodiment of the present invention;

FIG. 13 is an illustration showing a content of the data hold unit 1 in the migration communication control device employed as the gateway;

FIG. 14 is an illustration showing a content of the data hold unit 1 in the migration communication control device employed as the individual node;

FIG. 15 is an illustration showing an example of a network to which the migration communication control device is attached in a second embodiment of the present invention;

FIG. 16 is a detailed block diagram depicting a home migration communication control device in the second embodiment of the present invention;

FIG. 17 is an illustration showing a content of a home mobile host list hold unit in the second embodiment of the present invention;

FIG. 18 is a detailed block diagram depicting the visitor migration communication control device in the second embodiment of the present invention;

FIG. 19 is an illustration showing a content of a visitor mobile host list hold unit in the second embodiment of the present invention;

FIG. 20 is a detailed block diagram depicting a migration address unit in the second embodiment of the present invention;

FIG. 21 is an illustration showing a content of an address hold unit in the migration address unit in the second embodiment of the present invention;

FIG. 22 is a detailed block diagram depicting a migration address unit in the second embodiment of the present invention;

FIG. 23 is an illustration showing a content of the address hold unit in the migration address unit in the second embodiment of the present invention;

FIG. 24 is an illustration showing a format of a data packet in the second embodiment of the present invention;

FIG. 25 is an illustration showing a format of a packet transfer message in the second embodiment of the present invention;

FIG. 26 is an illustration showing a flow of a data packet transmitted between devices in the second embodiment of the present invention;

FIG. 27 is an illustration showing a communication sequence in FIG. 26;

FIG. 28 is an illustration showing a construction of each data packet in FIG. 26;

FIG. 29 is an illustration showing a change in the content of each hold unit in FIG. 26;

FIG. 30 is an illustration showing a flow of each data packet transmitted between devices at an operation example in the second embodiment of the present invention;

FIG. 31 is an illustration showing a communication sequence in FIG. 30;

FIG. 32 is an illustration showing a construction of each data packet in FIG. 30;

FIG. 33 is an illustration showing a change in the address hold unit in each device in FIG. 33;

FIG. 34 is an illustration showing a flow of a data packet transmitted between devices at an operation example in the second embodiment of the present invention;

FIG. 35 is an illustration showing the communication sequence in FIG. 34;

FIG. 36 is an illustration showing a construction of each data packet in FIG. 34;

FIG. 37 is an illustration showing a change in the address hold unit in each device in FIG. 34;

FIG. 38 is an illustration showing a flow of each data packet transmitted between devices at an operation example in the second embodiment of the present invention;

FIG. 39 is an illustration showing a communication sequence in FIG. 38;

FIG. 40 is an illustration showing a construction of each data packet in FIG. 38;

FIG. 41 is an illustration showing a change in the address hold unit in each device in FIG. 38;

FIG. 42 is an illustration showing a flow of each data packet transmitted between devices in the second embodiment of the present invention;

FIG. 43 is an illustration showing a flow of each data packet transmitted between devices in the second embodiment of the present invention;

FIG. 44 is an illustration showing a flow of each data packet transmitted between devices in the second embodiment of the present invention; and

FIG. 45 is an illustration showing a flow of each data packet transmitted between devices in the second embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

[Embodiment 1]

A construction of a migration communication control device in a first embodiment of the present invention is described hereunder with referring to FIGs. Hereinafter, the mobile node and partner node in the related art as well as in the summary of the invention are referred to as a mobile host and a stationary host, respectively.

FIG. 1 is an illustration showing the construction of the migration communication control device comprising a data hold unit 1, an application unit 2, a migration address unit 3, and a communication control unit 4.

The data hold unit 1 holds a couple of addresses of a mobile host by corresponding them. Each of the addresses in the data hold unit 1 is assigned before and after a migration of the mobile host.

The application unit 2 checks a connection as well as monitors a timer. The unit 2 is relevant for a higher layer in OSI model, which includes an application layer. For example, the unit 2 operates as TCP at TCP/IP (Transmission Control Protocol/Internet,Protocol) or a layer which is higher than TCP.

The migration address unit 3 processes a migration address with referring to data in the data hold unit 1. The concrete operation of the migration address unit 3 varies depending on a type of the migration communication control device comprising the unit 3, and this will be described in detail later.

The communication control unit 4 controls the communication. The unit 4 is relevant for a lower layer in the OSI model. For example, the unit 4 operates as a layer which is lower than IP at TCP.

The application unit 2 and the communication control unit 4 are the same units as ones implemented on a general host. Besides the unit 2 and 4, the migration communication device in the first embodiment of the present invention includes the data hold unit 1 and the migration address unit 3; thereby implements an operation unique to this case. That is, the data hold unit 1 and the migration address processing unit 3 are attached to the mobile host which migrates across networks, or a stationary host which is attached to a network fixedly (for example, a gateway or a server); otherwise, they operate alone. Each device comprising the unit 1 and 3 supports a continuous communication unaffected by migration of the mobile host besides providing its own function.

The data hold unit 1 and the mobile address unit 3, which are included in the devices attached to the network, are described in FIGs. 2, 3, 4, 5. FIG. 2 shows a migration communication control device where the unit 1 and the unit 3 are attached to the mobile host which migrates across networks; FIG. 3 shows a migration communication control device where the unit 1 and the unit 3 are attached to a gateway which connects the networks; FIG. 4 shows a migration communication control device where the unit 1 and the unit 3 are attached to the stationary host, which is the communication partner of the mobile host; and FIG. 5 shows a migration communication control device attached to the network itself.

The migration communication control device in FIG. 2 (hereinafter referred to as a mobile host) further includes the application unit 2, the communication control unit 4, and an address obtainment unit 25, besides the data hold unit 1 and the migration address unit 3.

Each of the application unit 2 and the communication control unit 4 operates as the above; while the unit 2 together with the unit 4 operate as a conventional stationary host.

The address obtainment unit 25 obtains an address of the mobile host assigned when it has migrated to another network. Although other options can be considered, such as employing a manual setting by an operator or communicating with a server computer which administrates addresses of the network, it is supposed here that the address is obtained in accordance with an instruction of a system administrator or the operator. The address obtainment unit 25 is also possessed by a general host and will not be described in detail.

The addresses held in the data hold unit 1 are obtained by the address obtainment unit 25.

The migration address unit 3 (enclosed with a broken line) consists of a response message transmission unit 20, a marked packet conversion unit 21, a migration address setting unit 26, a migration post transmission unit 27, a reception packet unit 28, and a marked packet resumption unit 29.

The response message transmission unit 20 transmits the packet which responds to the received packet if the response is needed.

The marked packet conversion unit 21 converts a packet received from the response message transmission unit 20 as well as the application unit 2 into a marked packet by converting the address of the received packet and marking the packet.

The migration address setting unit 26 stores the address obtained by the address obtainment unit 25 into the data hold unit 1. The address obtained by the unit 25 is the address of the mobile host assigned after the migration, and the unit 26 stores it into unit 1 by corresponding it to the address of the mobile host assigned before the migration.

The migration post transmission unit 27 posts via the communication control unit 4 that the address obtained by the unit 25 is held in the data hold unit 1 together with the correspondence between a couple addresses each of which assigned before and after the migration.

The reception packet unit 28 detects whether or not the received packet is marked, and sends the unmarked packet to the application unit 2 while sending the marked packet to the marketed packet resumption unit 29.

The marked packet resumption unit 29 resumes the marked packet.

The migration communication control device in FIG. 3 (hereinafter referred to as a gateway) further includes the application unit 2 and the communication control unit 4 besides the data hold unit 1 and the migration address unit 3 (enclosed with a broken line).

Each of the application unit 2 and the communication control unit 4 operates described the above, and the unit 2 together with the unit 4 operate as a conventional gateway.

The data hold unit 1 holds the correspondence between a couple of the addresses of the mobile host each of which assigned before and after migration.

The migration address unit 3 consists of a reception packet unit 35, a migration post information unit 36, an address comparison unit 37, an address conversion post transmission unit 38, and a marked packet conversion unit 39.

The reception packet unit 35 detects whether or not the received packet is the packet comprising a migration post message, which is transmitted by the mobile host. The unit 35 then sends the migration post message to the migration post information unit 36 while sending the other packets to the address comparison unit 37.

In accordance with the migration post message received from the reception packet unit 35, the migration post information unit 36 stores in the data hold unit 1 the correspondence between a couple of the addresses of the mobile host each of which assigned before and after the migration. The unit 36 also sends the migration post message to the address conversion post transmission unit 38.

The address comparison unit 37 detects whether or not the destination address of the packet received from the reception packet unit 35 coincides with the address of the mobile host assigned before migration, which is held in the data hold unit 1. When they coincide with each other, the unit 37 further sends to the marked packet conversion unit 39 the address assigned after the migration, which corresponds to the address which coincides with the destination address, as well as the packet received from the reception packet unit 35. On the other hand, when they do not coincided with each other, the unit 37 implements a function of a gateway by sending the packet to the application unit 2.

The address conversion post transmission unit 38 transmits to the destination address of the above packet received from the reception packet unit 35 an address conversion post message to inform that the address of the mobile host changes when the address comparison unit 37 detects a coincidence. Also the unit 38 transmits the address conversion post message to the network which satisfies the following two conditions: (1) the network where the address assigned before the migration, which is held in the data hold unit 1, is other than 0 (2) the migration communication control device employs as the gateway is not attached to the network. When the address conversion post message is transmitted to the network, which satisfies the above conditions, its destination address is a broadcast address of the network. The broadcast address consists of a network part and a host part, and every bit of the host part is 1.

The marked packet conversion unit 39 generates a marked packet when the address comparison unit 37 detects a coincidence. The unit 39 generates it by marking a general packet after converting the destination address of the packet. Then, the unit 39 transmits it.

The migration communication control device in FIG. 4 (hereinafter referred to as a stationary host) further includes the application unit 2 and the communication control unit 4 besides the data hold unit 1 and the migration address unit 3 (enclosed with a broken line).

Each of the application unit 2 and the communication control unit 4 operates as described the above, and the unit 2 together with the unit 4 operate as a conventional stationary host (not migrate).

The data hold unit 1 holds the correspondence between a couple of the addresses of the mobile host each of which assigned before and after the migration.

The migration address unit 3 consists of a reception packet unit 45, a marked packet resumption unit 46, an address conversion post information unit 47, an address comparison unit 48, and a marked packet conversion unit 49.

The reception packet unit 45 detects whether the received packet is the packet comprising the address conversion post message, the marked packet, or the other packets. The address conversion post message is transmitted by the gateway. Then the unit 45 sends the address conversion post message to the address conversion post information unit 47, the marked packet to the marked packet resumption unit 46, and the other packets to the application unit 2.

The marked packet resumption unit 46 resumes the unmarked packet from the marked packet, which is received from the reception packet unit 45.

The address conversion post information unit 47 obtains from the packet comprising the address conversions post message, which is received from the reception packet unit 45, the correspondence between the address of the mobile host assigned before the migration and the one assigned after the migration, and stores it into the data hold unit 1.

The address comparison unit 48 detects whether or not destination address of the packet received from the application unit 2 coincides with the address of the mobile host assigned before migration, which is held in the data hold unit 1. When they coincide with each other, the unit 48 further sends to the marked packet conversion unit 49 the address assigned after the migration, which corresponds to the address which coincides with the destination address, as well as the packet received from the application unit 2. On the other hand, when they do not coincided with each other, the unit 48 sends the packet to the communication control unit 4.

The marked packet conversion unit 49 generates a marked packet when the address comparison unit 37 detects a coincidence. The unit 49 generates it by marking a general packet after converting the destination address of the packet. Then, the unit 49 transmits it.

The migration communication control device in FIG. 5, which is attached to the network by itself, consists of the data hold unit 1, the migration address unit 3 (enclosed with a broken line), and the communication control unit 4.

The data hold unit 1 holds the correspondence between a couple of the addresses of the mobile host each of which assigned before and after the migration.

The migration address unit 3 consists of the reception packet unit 35, the migration post information unit 36, the address comparison unit 37, the address conversion post transmission unit 38, and the marked packet conversion unit 39. The units integrating the migration address unit 3 operate substantially same as equivalent units integrating the gateway in FIG. 3 except the following.

In FIG. 3 the address conversion post transmission unit 38 transmits the address conversion post message to the network satisfying both of the two conditions, which are described in the above; whereas, the address conversion post transmission unit 38 in FIG. 5 transmits the address conversion post message to the broadcast address of the network as long as the network satisfies the first condition, that is it transmits the address conversion post message to the network when the address assigned before the migration, which is held in the data hold unit 1, is other than 0.

FIG. 6 shows a first example of a network to which the migration communication control device as the mobile host in FIG. 2, the migration communication control device as the gateway in FIG. 3, and migration communication control device as the stationary host in FIG. 4 are attached. In the figure numeral 11 denotes a mobile host in FIG. 2, which migrates from a network A to a network B and obtains an address α assigned on the network A as well as an address β assigned on the network B.

Numeral 12 denotes a stationary host in FIG. 3, which is attached to the network B and obtains an address γ assigned thereon.

Numeral 12' denotes a stationary host in FIG. 3, which is attached to the network A and obtains an address γ' assigned thereon.

Numeral 13 denotes a gateway in FIG. 3, which has an address g. The gateway 13 is attached to both the network A and the network B.

The address on each network is assigned by a system administrator.

FIG. 7 shows a second example of a network to which the mobile host in FIG. 2, the gateway in FIG. 3, and the stationary host in FIG. 4 are attached. The stationary host is not illustrated in FIG. 7 since its location does not affect the communication with the mobile host.

In the figure the mobile host 11 migrates across network 1-4, and obtains an address m, m', m'', m''' assigned on each network respectively.

The network 5 as well as each of the network 1-4 (hereinafter referred to as the net 5, and the net 1-4 respectively) are connected with each other by a gateway 1-4, as shown in the figure.

A gateway 1-4 (hereinafter referred to as gw 1-gw 4) is the migration communication control device employed as the gateway in FIG. 3.

FIG. 8 shows a third example of the network to which the mobile host in FIG. 2, the gateway in FIG. 3, and the stationary host in FIG. 4 are attached. Construction of this network is substantially same as the second example of the network in FIG. 6 although operation thereof is different from the second example, which will be described later.

FIG. 9 shows a fourth example of the network to which the mobile host in FIG. 2, the migration communication control device in FIG. 5, the stationary host in FIG. 4 are attached. The migration communication control device as the stationary host will not be described here.

In the figure, numeral 11 denotes the mobile host which migrates across the network 1-4 and obtains the address m, m', m", m'" assigned on each network respectively.

The network 5 as well as each of the network 1-4 (hereinafter referred to as the net 5, and the net 1-4 respectively) are connected with each other by a gw 1-4, as shown in the figure.

Each of the migration communication control unit 1-4 (hereinafter referred to as S1-S4) is relevant for the one in the FIG. 5.

An address used in the first embodiment of the present invention is described hereunder. Each address consists of a network part, which is assigned on each network and shared by every host attached to that network, as well as a host part, which is assigned to each host uniquely.

A broadcast address is a special kind of address, which can be divided into two types. The first one is the broadcast address used as the destination address in transmitting a packet from a network to another network, such as the broadcast address where every bit of the host part is 1. When the first type of the broadcast address is used as the destination address of the packet, the packet is transferred by the gateway to the network directed by the network part of the broadcast address. The other one is used in transmitting a packet within a network, such as the broadcast address where every bit of both the host part and the network part is 1. When the second type of the broadcast address is used as the destination address of the packet, the packet is transmitted to all the devices attached to the network, which includes the broadcast address. However, the gateway does not transfer the packet to any other network.

Operations of the migration communication control device in the first embodiment of the present invention are described hereunder with referring to drawings.

(operation example in FIG. 6)

In FIG. 6, when the mobile host migrates from the network A to the network B, the migration communication control device is operated as follows.

In a first operation, the mobile host and the gateway operate when the mobile host migrates across networks.

In a second operation, the stationary host transmits a packet to an address of the mobile host which was assigned before the migration.

In a third operation, the stationary host transmits the packet to an address of the mobile host which has been assigned after the migration.

In a fourth operation, the mobile host receives the packet which is transmitted by the stationary host.

In a fifth operation, the mobile host sends a response message to the stationary host.

(first operation in FIG. 6)

In FIG. 6 the mobile host 11 attached to the network A (enclosed with a broken line) migrates to the network B to complete ongoing communication with the stationary host 12, which is attached to the network B. When migrating to the network B, the address obtainment unit 25 in the mobile host 11 (FIG. 2) obtains the address β assigned on the network B.

Immediately after obtaining the address β, the address obtainment unit 25 gives the address β to the migration address setting unit 26 and the migration post transmission unit 27. The migration address setting unit 26 stores the address β into the data hold unit 1 by corresponding it to the address α, which is the address assigned before the migration. FIG. 10 (a) shows the content of the data hold unit 1. The migration post transmission unit 27 gives to the gateway 13 via the communication control unit 4 a packet comprising migration post message and the correspondence between the address α and the address β, so that the gateway 13 will know that the mobile host 11 has migrated to the network B. The mobile host 11 can transmit the packet both before and after the migration. In FIG. 6 a packet 51 is transmitted before the migration, and its format is shown in FIG. 11 (a). As shown in FIG. 11 (a), the packet 51 consists of a destination address 91, a source address 92, and data 93. The data 93 further comprise a message type 98, an address before migration 94, and an address after migration 95.

Receiving from the communication control unit 4 the packet 51, the gate way 13 sends it to the reception packet unit 35, the unit 4 and the unit 35 being in FIG.:3. From the message type 98 in FIG. 11 (a), the gateway 13 identifies the packet 51 with the migration post message, and gives the packet 51 to the migration post information unit 36. The migration post information unit 36 obtains from the data 93 in the data packet 51 the address before migration α and the address after the migration β; then stores them into the data hold unit 1 by corresponding them with each other. The content of the data in the data hold unit 1 is shown in FIG. 10 (b).

Additionally, the destination address 91 of the packet in FIG. 11 (a), can be the broadcast address of the network A, where the network part names the network A and every bit of the host part is 1. When the broadcast address is employed, every stationary host attached to the network A, including the gate way 13, receives the correspondence of the addresses each of which assigned before and after the migration. In this case communication control unit 4 in the stationary host 12' receives the data packet 51, and gives it to the reception packet unit 45, the unit 4 and the unit 45 in FIG. 4. From the message type 98 in FIG. 11 (a), the reception packet unit 45 identifies the packet 51 with the migration post message, and gives the packet 51 to the address conversion post information unit 47. The unit 47 obtains from the data 93 in the data packet 51 the address before migration α and the address after the migration β and stores them into the data hold unit 1 by corresponding them with each other. Once those addresses are stored in the data hold unit 1, the stationary host 12' can transmit a packet to the address assigned after the migration instead of transmitting it to the address before the migration, the same to other stationary hosts attached to the network A.

(second operation in FIG. 6)

In the second operation, the stationary host 12 transmits a packet to the address assigned before the migration after the mobile host 11 migrates to the network B and obtains the address β assigned on the network B. It is supposed that the mobile host 11 transmits the packet 51, which comprises the migration post message, to the gateway 13 rather than to the broadcast address of the network A.

The stationary host 12, which is not notified that the mobile host 11 has migrated to the network B, transmits the packet to the address α of the mobile host, which was assigned before the migration. A packet 52 in FIG. 6 is transmitted by the stationary host 12 to the address α of the mobile host 11, and its format is shown in FIG. 11 (f). The packet 52 is received by the gateway 13. Because the gateway 13 is located between the source address of the packet 52 and the address of the mobile host α assigned before the migration, and also it is attached to the network A, to which the mobile host 11 was attached before the migration.

The gateway 13 employs its devices in FIG. 3 to implement its functions including reception of the packet. That is, the communication control unit 4 in the gateway 13 receives the packet 52, and sends it to the reception packet unit 35 in the migration address unit 3. The reception packet unit 35 identifies the packet 52 with a general packet and gives it to the address comparison unit 37. The unit 37 compares the destination address α of the packet 52 with the address before the migration, which is held in the data hold unit 1; then detects whether or not they are coincide with each other. When the destination address of the received packet does not coincide with the address assigned before the migration, the address comparison unit 37 sends the packet to the application unit 2. On the other hand, when they coincide with each other, the address comparison unit 37 obtains from the data hold unit 1 the address β of the mobile host assigned after the migration, which corresponds to the address α; then sends it both to the address conversion post transmission unit 38 and the marked packet conversion unit 39.

As is described the above, the packet 52 is transmitted to the address α of the mobile host 11 by the stationary host 12. Therefore, the address conversion post transmission unit 38 notifies the stationary host 12 that the address of the mobile host 11 has changed by transmitting thereto the packet 53. FIG. 11 (b) shows the packet 53. Simultaneously, the marked packet conversion unit 39 converts the packet 52 into the packet 53 by rewriting the destination address of the packet 52 to the address β assigned after the migration, returning thereto the previous destination address of the packet 52 as additional information, and marking to show that its destination address has changed; then sends the packet to the communication control unit 4. Thereby, the packet 52, which is converted into the marked packet 52', is transferred from the address α of the mobile host 11 assigned before the migration to the address β assigned after the migration. FIG. 12 (e) shows the packet 52'.

Receiving the packet 53 from the communication control unit 4 in the stationary host 12, it sends its packet 53 to the reception packet unit 45, the unit 4 and the unit 45 being in FIG. 4. From the message type 98 in FIG. 11 (b), the reception packet unit 45 identifies the packet 53 with the address conversion post message, and gives the packet 53 to the address conversion post information unit 47. The address conversion post information unit 47 obtains from the data 93 in the data packet 53 the address before migration α and the address after the migration β; then stores them into the data hold unit 1 by corresponding them with each other. Thereby, the stationary host 12 obtains the address of the mobile host 11 assigned after the migration, so that a direct communication between the stationary host 12 and the mobile host 11 is implemented.

In the second operation the migration communication control device comprising the units in FIG. 4 is employed as the stationary host 12. However, a conventional stationary host, which is not constructed as the migration communication control device can also be communication partner of the mobile host if it is attached to a network. Therefore, hereunder a communication between the mobile host 11 and the convention stationary host is described.

When the conventional stationary host transmits a packet to the address of the mobile host 11 assigned before the migration after the mobile host 11 has migrated to another network, the gateway 13 transfers the packet to the address of the mobile host 11 assigned after the migration as well as sends to the stationary host the packet 53 comprising the address conversion post message in FIG. 11 (c). This operation is same as the above.

However, when receiving the packet 53, the stationary host disposes it since it does not support the address conversion post message and judges the packet 53 is not a required packet. Thus, the conventional stationary host cannot utilize the packet 53 to detect the address of the mobile host assigned after the migration nor hold the correspondence of the addresses each assigned before and after the migration.

Therefore, the stationary host gives the packet only to the address of the mobile host 11 assigned before the migration. Then, the gateway transfers the packet to the address of the mobile host 11 assigned after the migration, and the mobile host 11 receives the packet. The message from the mobile host 11, such as the response message, is transmitted to the stationary host directly, so that it is received by the stationary host without fail.

Thus, the conventional stationary host transmits a packet to the mobile host indirectly and receives a packet from the mobile host directly. Continuous communication unaffected by the mobile host's migration can be implemented, even when the conventional stationary host is employed.

(third operation in FIG. 6)

In the third operation, the stationary host 12 transmits the packet to the address β of the mobile host 11 assigned after the migration with referring to the correspondence of the addresses each assigned before and after the migration, which is held in the data hold unit 1. The third operation is described hereunder with referring to FIG. 4.

The stationary host 12 employs its devices in FIG. 4 to implement conversion of the destination address and the transmission of the packet, both of which integrate the third operation. That is, application unit 2 sends to the address comparison unit 48 the packet 54, whose destination address is the address α of the mobile host 11 assigned before the migration. FIG. 11 (f) shows a format of the packet 54. Then, the comparison unit 48 obtains the destination address of the packet 54 and detects whether or not it coincides with the address before the migration, which is held in the data hold unit 1.

The comparison unit 48 sends the packet 54 to the communication control unit 4 when the above addresses do not coincide with each other while it sends the packet 54 to the marked packet conversion unit 49 when the above addresses coincide with each other. In the third operation the coincidence is detected since the corresponded between the address α and the address β is stored in the data hold unit 1. Therefore, the packet 54 is sent to the marked packet conversion unit 49. Then the marked packet conversion unit 49 obtains from the data hold unit 1 the address β of the mobile host assigned after the migration, which corresponds to the address α as well as converts the packet 54 into the packet 54' by converting the destination address α into the address β, returning thereto the original destination address α as additional information, and marking the packet 54 to show that its destination address has changed; then sends the packet 54' to the communication control unit 4. FIG. 11 (c) shows a format of the packet 54'. Since the destination address of the packet 54' is an updated address of the mobile host 11, the packet 54' is given to the mobile host 11 without fail.

(fourth operation in FIG. 6)

In the fourth operation, the mobile host 11 receives the marked packet 54' and obtains the original unmarked packet 54 by resuming the packet 54'. This operation is described hereunder with referring to FIG. 2.

The mobile host 11 employs its devices in FIG. 2 to implement its operation. That is, the communication control unit 4 receives the packet 54' and sends it to the reception packet unit 28. The reception packet unit 28 detects that the received packet 54' is marked, and sends it to the marked packet resumption unit 29. The unit 29 obtains the original destination address α, which is held in the additional information 97, and replaces the current destination address β of the packet 54' with the address α. Then it sends the packet 54' to the application unit 2. Thus, the mobile host 11 can receive the packet destined for its outdated address.

(fifth operation in FIG. 6)

In the fifth operation, the mobile host 11 sends to the stationary host 12 a packet comprising a response message (hereinafter referred to as a response packet) or a packet excluding the response message (hereinafter referred to as a non-response packet). A type of the received packet determines whether or not it is responded with the response packet.

When the packet 54' is responded with a response packet, the mobile host 11 employs its devices in FIG. 2 to send the response packet. That is, the response message transmission unit 20 builds the response packet, and sends it to the marked packet conversion unit 21 together with the destination address α of the packet 54'.

The mobile host 12 also employs its devices to send the non-response packet 55. That is, the application unit 2 gives the address α assigned before the migration and the non-response packet to the marked packet conversion unit 21. The unit 21 sends the received packet to the stationary host 12 via the communication control unit 4 without marking it. FIG. 11 (e) shows the packet sent by the unit 21 to the stationary host 12.

The communication control unit 4 in the stationary host 4 receives the packet 55, and gives it to the reception packet unit 45. The unit 45 detects that the packet 55 is the non-response packet, so that it gives the packet 55 to the application unit 2. Thus, the stationary host and the mobile host implement a continuous communication unaffected by mobile host's migration. Although the migration communication control device is employed as the stationary host 12 in this embodiment, the conventional host can also be employed to transmit the non-response packet.

In the above, the unmarked response packet and the unmarked non-response packet are sent to the mobile stationary host 12. On the other hand, hereunder the operation of the mobile host 11 at conversion of the response packet and the non-response packet into the marked ones is described. This will be employed effectively in a communication between mobile hosts.

Receiving the unmarked packet from the application unit 2, the marked packet conversion unit 21 generates a packet 55' where the destination address and the source address are the address γ of the stationary host 12 and the address β assigned after the migration respectively. Also in generating the packet 55', the application unit 2 gives to the received packet the address α assigned before the migration as additional information as well as marks the received packet to indicate that the destination address has converted. FIG. 11 (d) shows a format of the packet 55'. Then the application unit 2 sends the packet 55' to the stationary host 12 via the communication control unit 4.

The communication control unit 4 in the stationary host 12 receives the packet 55', and sends it to the reception packet unit 45. Detecting the packet 55' is the marked packet, the reception packet unit 45 sends it to the marked packet resumption unit 46. The unit 46 resumes the packet 55' into the packet 55 by unmarking it and replacing the source address thereof with the address α assigned before the migration, which is held as the additional information. A format of the packet 55 is shown in FIG. 11 (e). Thus, the stationary host and the mobile host implement a continuous communication unaffected by mobile host's migration.

(operation example in FIG. 7)

In FIG. 7, when the mobile host migrates across the network 1, 2, 3, and 4, and obtains a temporary address assigned on each network, the newest address of the mobile host is transmitted to the stationary host, which operates as communication partner.

(migration from network 1 to network 2)

The address of the mobile host is m when it is attached to the network 1. When migrating from the network 1 to the network 2, the mobile host 11 replaces its address with m' assigned on the network 2. Then the mobile host 11 notifies the migration communication control device attached to the network 1 that it has migrated to the network 2 by sending thereto a packet comprising a migration post message. In FIG. 7 the migration communication control device gw 1, gw 2 attached to the network 1 receive the migration post packet 61, and store it into its own data hold unit 1. The operation in FIG. 7 is substantially same as the operation in FIG. 6 except that in FIG. 7 the packet 61 holds the address of the mobile host assigned before the last migration besides the correspondence of the addresses each assigned before and after the current migration. The address assigned before the last migration makes the gws prepare for further migration of the mobile host, which will be described later. A format of the packet 61 is shown in FIG. 12 (a). Since the migration from the network 1 to the network 2 is the first migration in FIG. 7, the packet 61 holds 0 at the address assigned before the last migration.

The gw 1 and the gw 2 store in the data hold unit 1 the correspondence of the addresses each assigned before and after the migration, as well as the address assigned before the last migration. As shown in FIG. 13 (a), m-m' and 0 are stored in the data hold unit 1 of each of the gw 1 and the gw 2.

Then, the gw 1 and the gw 2 detects from 0 at the address assigned before the last migration that no migration had been conducted before the current migration.

The broadcast address of the network 1 can be employed as the destination address of the migration post packet 61. If the packet is destined for the broadcast address, every host attached to the network 1, which includes the gw 1 and the gw 2, will hold the correspondence of the addresses each of which assigned before and after the migration as well as the address assigned before the last migration. Thereby, the hosts attached to the network 1 can communicated with the mobile host directly.

(migration from network 2 to network 3)

When migrating from the network 2 to the network 3, the mobile host 11 obtains m" at the address assigned after the migration. Then the mobile host 11 notifies the gw 2 and a gw 3, both of which are attached to the network 2, that the mobile host 11 has migrated to the network 3 by transmitting thereto a packet comprising the migration post message, referred to as a packet 62 in FIG. 7. FIG. 12 (b) shows a format of the packet 62, which is transmitted to the gw 2. The broadcast address of the network 2 can be employed as the destination address of the packet 62. When the packet 62 is transmitted to the broadcast address of the network 2, every host attached to the network 2, which includes the gw 2 and the gw 3, holds the correspondence of the addresses each assigned before and after the migration.

The gw 2 employs its devices in FIG. 3 to process the packet 62. That is, receiving the packet 62, the gw 2 sends it to the migration post information unit 36 via the communication control unit 4 and the reception packet unit 35, then refers to the data hold unit 1 where m→m' and 0 are still held at the address correspondence and at the address assigned before the last migration respectively. The migration post information post unit 36 obtains from the packet 62 m'-m" as the newly assigned correspondence between the addresses each of which assigned before and after the current migration, the migration from the network 2 to the network 3. Then, it detects whether or not the address m' coincides with the address held in the data hold unit 1 as the address assigned after the last migration. Since the unit 36 detects the coincidence, it replaces the address m' in the unit 1 with the address m" as well as replaces the correspondence m-m' with the correspondence m-m".

Also the migration post information unit 36 sends to the data hold unit 1 the address m assigned before the last migration together with the address correspondence m'-m" obtained from the current migration. Now the data hold unit 1 in the gw 2 holds the address m at the address assigned before the last migration and the address correspondence m'-m" at the correspondence of the addresses each of which assigned before and after the migration as well as the address 0 at the address assigned before the last migration as well as the address correspondence m-m' at the correspondence of the addresses each of which assigned before and after the migration. After updating as well as adding the addresses in the data hold unit 1, the migration post information unit 36 sends to the address conversions post transmission unit 38 m'-m" as the newly obtained correspondence of the addresses before and after the current migration.

The address conversion post transmission unit 38 detects the network satisfying the following conditions with referring to the data hold unit 1 and then transmits the address conversion post message to the broadcast address of the detected network. That is, the address conversion post message is transmitted to the network where the address assigned before the migration, which is held in the data hold unit 1, is other than 0 as well as the migration communication control device employed as the gateway is not attached. Although in the migration from the network 2 to the network 3, the data hold unit 1 holds m at the address assigned before the last migration, the gw 2 is attached to the network 1; therefore, the unit 38 does not transmit the address conversion post to the network 1.

The packet 62 is also received by gw 3. When receiving the packet 62, the gw 3 employs its own devices in FIG. 3 to process the packet 62, which is substantially same as does the gw 2 except the following. That is, the address conversion post transmission unit 38 of the gw 3 detects that tne gw 3 is not attached to the network 1. Also it is detected that the mobile host 11, attached to the network 1, has the address m as the address assigned before the last migration. Therefore, the unit 38 of the gw 3 transmits to the broadcast address of the network 1 a packet comprising the address conversion post message, which is referred to as a packet 63. FIG. 12 (c) shows the packet 63.

The packet 63 is received by the gw 2, the gw 1, both of which are attached to the network 1. Although it is also received by the stationary host 11, this will not be described here. Obtaining the current address correspondence m'-m" from the packet 63, where m' coincides with the address which has been held in the hold unit 1 at the address obtained after the migration, the gw 1 changes the m-m' in the data hold unit 1 into the m-m" by replacing m' with m" as the address assigned after migration.

On the other hand, the data hold unit 1 of the gw 2 had gained from the packet 62 the above information before receiving the packet 63. Therefore the content of the unit 1 of the gw 2 does not change across reception of the packet 63. This is because the gws of the present invention locate on a gateway, which connects a couple of networks. Due to its location, each gw receives packets from two networks. However, actually the packet 62 is destined for the network 2 and the packet 63 is destined for the network 1. Therefore, even though the gw 2, which are attached to both the network 1 and the network 2, receives both the packet 62 and 63 by the gw 2, this will not cause any problem in the communication between the stationary host 12 and the mobile host 11.

FIG. 13 (b) shows the content of the data hold unit 1 in each of the gws.

(migration from network 3 to network 4)

When migrating from the network 3 to the network 4, the mobile host 11 obtains m''' as the address assigned after the migration. Then the mobile host 11 sends to the gw 3 and a gw 4, both of which are attached to the network 3, a packet comprising the migration post message. The packet received by the gw 3 is referred to as a packet 64. The broadcast address of the network 3 can be employed as the destination address of the packet 64. When the packet 64 is destined for the broadcast address of the network 3, every host attached to the network 2, which includes the gw 3 and the gw 4, obtains from the packet the correspondence of the addresses each of which assigned before and after the migration from the network 3 to the network 4.

The gw 3 employs its devices in FIG. 3 to process the packet 34. That is, receiving the packet 64, the gw 3 converts the content of the data hold unit 1 by replacing the address correspondence m-m'' with m-m''', newly holding m''-m''' obtained from the packet 64 as well as the address m' assigned before the last migration. Then, the address conversion post transmission unit 38 of the gw 3 transmits the address conversion post message to the network satisfying the following condition. That is, the address conversion post message is transmitted to the network where the address assigned before the migration, which is held in the data hold unit 1, is other than 0 as well as the gw 3 it self is not attached. The packet including the address conversion post message is referred to a packet 65, and the packet is transmitted to the broadcast address of the network 1. FIG. 7 (c) shows the packet 65.

The packet 64 is also received by gw 4. When receiving the packet 64, the gw 4 renews the content of the data hold unit 1 by replacing m'-m'' with m'-m''' as well as newly holding the address m' as the address assigned before the last migration. Further, the address conversion post transmission unit 38 of the gw 4 detects that the gw 4 is not attached to the network 2 which has the address other than 0 at the address assigned before the last migration; therefore, the unit 38 of the gw 4 transmits a packet comprising the address conversion post message, which is referred to as a packet 66, to the broadcast address of the network 2. FIG. 7 (c) shows the packet 66.

Receiving the packet 65, 65, the gw 2 and the gw 1 renew the content of its data hold unit 1, which is substantially the same as the above.

The gw 3 and the gw 2 receives the same information twice since the former receives the packet 64 and 65 while the latter receives the packet 65 and 66. This is because gws of the present invention locate on a gateway and receives packets from a couple of networks, which is described the above.

FIG. 13 (c) shows the content of the data hold unit 1 in each of the gws. Thus, according to the gws of the present invention, the packet transmitted to any of the addresses m, m', m'' is transferred by the gws to the updated address of the mobile host, the gws also notify the stationary host of the updated address.

For example, when the stationary host is not notified of the updated address of the mobile host and transmits a packet to the address m', the packet is received by the gw 2 and the gw 3, both of which are attached to the network 2. Then, the gw 2 and the gw 3 transfers the packet to the updated address of the mobile host as well as notifies the stationary host of the updated address. Thereby, the stationary host obtains the updated address of the mobile host, so that it will be able to communicate with the mobile host directly. The packet destined for the address m' is received by both the gw 2 and the gw 3, since they are attached to the network 2. Thus, the mobile host receives the same packet twice, once from the gw 2 and the other time from the gw 3, and the stationary host receives the same message twice; however, the repeated packet or the message can be simply ignored, so that this will not cause any problem in the communication between the stationary host and the mobile host. The repeated packet or the message is observed when the two gws are attached to each network in FIG. 7; whereas it is not observed when only one migration communication control device is attached to each network, which will be described later at the operation in FIG. 9.

(operation example in FIG. 8)

In FIG. 6, FIG. 7, the stationary host transmits the data packet to the outdated address after mobile host notifies the gws that it has migrated to another network. Then the gws transmit the address conversion post message to the stationary host. However, in FIG. 8 the gws convert the destination address of data the packet from the outdated address into the updated address assigned after the migration instead of transmitting the address conversion post message.

A packet 71, 72 in FIG. 8 are substantially same as the packet 51, 52 in FIG. 6. The operation conducted before the packet 72 is transmitted by the stationary host 12 and is received by the gateway 13 is substantially same as the first operation in FIG. 6. The operation which follows reception of the packet 72 is described hereunder with referring to FIG. 3.

The gate way 13 employs its units in FIG. 3 to process the packet 72. The communication control unit 4 receives the packet 72 and gives it to the reception packet unit 35 in the migration address unit 3. Detecting that the packet 72 is a general packet, the reception packet unit 35 sends it to the address comparison unit 37. The address comparison unit 37 detects whether or not the destination address of the packet 72 coincides with the address in the data hold unit 1 at the address assigned before the migration.

When no coincides is found, the address comparison unit 37 gives the packet 72 to the application unit 2. On the other hand, a coincidence is found, the address assigned after the migration, which corresponds with the address identical to the destination address of the packet 72, is obtained from the data hold unit 1, and is sent to the marked packet conversion unit 39 together with the packet 72. The marked packet conversion unit 39 generates a packet 72' where the destination address of the packet 72 is replaced with the address assigned after the migration, which is sent by the address comparison unit 37, the destination address of the packet 72 is added as additional address, and a mark is set to indicate that the destination address has converted. Then the packet 72' is sent to the communication control unit 4. FIG. 12 (e) shows a format of the packet 72', where identical numerals denotes the same units in FIG. 11. The packet 72' is sent to the mobile host 11 without fail since its destination address is the updated address thereof.

(operation example in FIG. 9)

In FIG. 9, the mobile host migrates across network 1, 2, 3, and 4. In FIG. 7 the gw 1-gw 4 are employed as the migration communication control devices; whereas in FIG. 9 the gw 1-gw 4 are employed simply as gateways to connect networks, and also another migration communication control device is attached to each network. The operation of the migration communication control device, which is connected to the network alone, at processing the migration post message or the address conversion post message is substantially same as one of the gw 1-gw 4 in FIG. 7. The flow of the migration post message and the address migration post message are mainly described hereunder.

(migration from network 1 to network 2)

When migrating from the network 1 to the network 2, the mobile host 11 sends a packet comprising the migration post message to the migration communication control device, which is attached to the network 1. In FIG. 9 (a) a migration post packet 81 is transmitted to a migration communication control device S1, which is attached to the network 1. The destination address of the packet 81 can be the broadcast address of the network 1.

The device S1 processes the packet 81 by employing its devices in FIG. 3. Receiving the packet 81, the device S1 stores into the data hold unit 1 the correspondence of the addresses each assigned before arid after the migration as well as the address assigned before the last migration. The migration post information unit 36 transmits the packet 81 to the address conversion post transmission unit 38; however, since the unit 38 detects that the address assigned before the last migration is 0, it does not transmit the address conversion post message to any network. The content of the data hold unit 1 in the S1-S4 are shown in FIG. 14 (a).

(migration from network 2 to network 3)

When migrating from the network 2 to the network 3, the mobile host 11 notifies the S2, which is attached to the network 2, that it has migrated to the network 3 by transmitting thereto the packet comprising the migration post message, which is referred to as a packet 82 in FIG. 9 (b).

The S2 employs its devices in FIG. 3 to process the packet 82. That is, it converts the content of the data hold unit 1 by renewing and adding new information, and finally holds in the unit 1 the address m'-m" at the correspondence of the addresses each of which assigned before and after the migration as well as the address m assigned before the last migration. Then, the migration post information unit 36 gives the newly obtained correspondence m'-m" to the address conversion post transmission unit 38.

The address conversion post transmission unit 38 detects whether or not the address assigned before the last migration, which is held in the data hold unit 1, is 0. If the address is not 0, the unit 38 transmits the address conversion post message to the broadcast address of the network which includes the detected address. In FIG. 9 (b) the address m is held at the address assigned before the last migration, so that the unit 38 transmits the packet 83 to the broadcast address of the network 1.

When receiving the packet 83, the migration communication control device S1, which is attached to the network 1, renews the content of the data hold unit 1 by newly holding the address correspondence m-m" as well as the address 0 at the address assigned before the last migration. Detecting 0 at the address assigned before the last migration, the address conversion post transmission unit 38 does not transmit the address conversion post to any network. The content of the data hold unit 1 in the S1-S4 are shown in FIG. 14 (b).

(migration from network 3 to network 4)

When migrating from the network 3 to the network 4, the mobile host 11 notifies the communication migration control device S3, which is attached to the network 3, that it has migrated to the network 4 by transmitting thereto a packet comprising the migration post message, referred to as a packet 84 ia FIG. 9 (c).

The migration communication control device S3 employs its devices in FIG. 3 to process the packet 84. That is, it newly holds into the data hold unit 1 the address correspondence m"-m'" as well as the address m' assigned before the last migration. Then, the address conversion post transmission unit 38 in the S3 transmits a packet comprising the address conversion post message, referred to a packet 85 in FIG. 9 (c), to the broadcast address of the network 2 since the address m' is held at the address assigned before the last migration in the data host unit 1.

When receiving the packet 85, the migration communication control device S2 employs its devices in FIG. 3 to process it. That is, it newly holds into the data hold unit 1 the address correspondence m'-m" as well as the address m assigned before the last migration. Then, the address conversion post transmission unit 38 in the S2 transmits a packet comprising the address conversion post message, referred to a packet 86 in FIG. 9 (c), to the broadcast address of the network 2 since the address m is held at the address assigned before the last migration in the data hold unit 1.

When receiving the packet 86, the migration communication control device S1 employs its devices in FIG. 3 to process it. That is, it newly holds into the data hold unit 1 the address correspondence m-m'" as well as the address 0 at the address assigned before the last migration. The address conversion post transmission unit 38 in the S1 does transmit the address conversion post since 0 is detected at the address assigned before the last migration. The content of the data hold unit 1 in each of the S1-S4 are shown in FIG. 14 (c). Thus, according to the migration communication control device S1-S4 of the present invention, the S1-S4 are notified of the updated address of the mobile host at every migration, so that the packet transmitted to any of the addresses m, m', m" is transferred thereby to the updated address of the mobile host. The S1-S4 also notify the stationary host of the updated address of the mobile host.

The operation in FIG. 9 differs from the operation in FIG. 7 in that each network has just one communication migration control device (one of the S1-S4), so that the migration post and the address conversion transmitted to S1-S4 are not duplicated.

In the format shown in FIG. 11 and 12, the mark 96 or the message type 93 indicates kind of packet. That is, mark 96 indicates whether or not the packet is marked while the message type 93 indicates whether it is the packet comprising the migration post message, the packet comprising the address conversion post message, and the general packet. Further, a protocol type can also be employed to indicate which migration communication control device is employed. For example, when TCP/IP is employed, the protocol number at the IP header thereof distinguishes the packet employed in the embodiment from other packets. That is, when the protocol number in the packet is identical with the one, which has been assigned to the protocol number field, the packet is the one employed in the embodiment.

In the first embodiment of the present invention, a nonvolatile storage can be employed as the data hold unit 1 of the mobile host. If so, the communication can be resumed even after the host or the gateway is turned off as well as after the system is reset.

Also even when the stationary host employs the nonvolatile storage as the data hold unit 1, it can resume the communication, which has interrupted by the switch off or the system reset, rather fast since it obtains from another host the updated address of the mobile host instead of receiving from the gateway the address conversion post message which shows the updated address.

For example, it is supposed in FIG. 7 that the mobile host 11 migrates from the network 1 to the network 4. The data hold unit 1 of the migration communication device holds the address correspondence m-m'" since it has communicated with the mobile host, which is attached to the network 4, at least once. According to the migration communication control device in the embodiment described the above, the packet is transferred from the outdated address to the updated address of the mobile host and the stationary host is notified of the updated address; therefore, even when the address information in the data hold unit is lost by switch off thereof, the stationary host will obtain the updated address. Restart of the communication can also be implemented by employing a specific host such as a server. That is, the server may be constructed to obtain the updated address of the mobile host at every migration, and give it to the stationary host whenever requested. In this case a packet comprising the address inquiry should be generated beforehand.

Also in the fifth operation in FIG. 6, the mobile host 11 employs the application unit 2 and sends to the marked packet conversion unit 21 the address assigned before the migration when transmitting the non-response address to the stationary host after it has migrated to another network. Instead of sending the non-response address, the application unit 2 can transmit a connection identifier to the marked packet conversion unit 21. In this case the data hold unit of the migration communication control device, employed as the mobile host, holds a correspondence between the connection identifier and the address that had been assigned when the connection was established instead of holding the correspondence between the correspondence of the addresses each assigned before and after the migration. Then, the unit 21 obtains the source address of the packet by detecting the address which corresponds to the identifier, which is held in the data hold unit 1.

As is described the above, the mobile host can employ the broadcast address of the network when transmitting the migration post to the migration communication control devices. When the broadcast address is employed, every host attached to the network, to which the migration communication control device is also attached, obtains the updated address of the mobile host. This implements a direct communication between the mobile host and the stationary host, which improves efficiency of the communication.

The address assigned before the last migration, which is held in the hold unit 1, can be replaced with the broadcast address assigned to the network to which the mobile host is attached before the last migration. If the broadcast address is employed, the gateway employed as the migration communication control device (gws) or the migration communication control device (Ss) needs to include the broadcast address in the address conversion post message. In this case both devices can obtain the broadcast address from the data hold unit; therefore, the operation thereof at requesting the broadcast address will be eliminated.

When storage capacity of the data hold unit 1 is limited, the data hold unit 1 holds only the useful data by disposing the unuseful data, which is least recently retrieved therefrom by the address comparison unit.

[Embodiment 2]

In FIG. 15 network A, B, and C are connected in a line via gateways 143 and 143', the gateway 143 placing between the network A and B while the gateway 143' placing between the network B and C.

A home migration communication control device 101 including a migration address unit 144 is attached to the network A; a visitor migration communication control device 109 including a migration address unit 145 is attached to the network B; and a visitor migration communication control device 109' including a migration address unit 145' is attached to the network C. A mobile host 146 including a migration address unit 115 is attached to the network A as its home network, and a stationary host 151 including a migration address unit 125 is also attached to the network A.

The mobile host 146 migrates across the network A, B, and C. It has a home address α assigned when it is attached to the network A, as well as other addresses assigned depending on where it migrates, such as a temporary address β on the network B and a temporary address γ on the network C.

Also each of the home migration communication control device 101, the visitor migration communication control device 109, 109' which are identical in its construction and the stationary host 151 has an address Ha, Va, Va', and Sa respectively assigned on the network.

Detailed function of the above devices 101, 109, 109', 146, and 151 is described hereunder, in which like components are labeled with like reference numerals.

[home migration communication control device 101]

When the mobile host 146 migrates from the home network to another network, it is assigned the temporary address. However if the stationary host 151 is not notified of that migration, it transmits an original data packet (hereinafter referred to as a noncapsulated data packet) to the home address α of the mobile host 146. When the noncapsulated data packet is destined for the outdated address of the home mobile host 146, the home migration communication control device 101 transfers that noncapsulated data packet from there to the updated address, that is the temporary address β or γ of the mobile host. Then, the device 101 posts to the stationary host 151 the temporary address β or γ here, so that the stationary host 151 will be able to communicate directly with the mobile host. The device 101 also posts the same information to the visitor migration communication control device 109, 109', so that the devices 109, 109' will implement the same function with the home migration communication control device 101.

As shown in FIG. 16 the home migration communication control device 101 consists of the migration address unit 144 and a communication control unit 108. The migration address unit 144 further comprises a home mobile host (MH) list hold unit 102, a packet transfer unit 103, a mobile host (MH) transfer unit 104, an address inquiry unit 105, a packet monitoring unit 106, an address post unit 107.

Next the function of each component integrating the device 101 will be described. The communication control unit 108 mainly controls the communication of protocols located in lower layers including a physical layer, such as the protocol lower than IP.

The address post unit 107 receives from the mobile host 146 an data packet including an address post message. The address post message is generated when the mobile host 146 migrates to the network B or C, and posts the temporary address β or γ of the mobile host to the device 101. The unit 107 sends the address post message to the mobile host transfer unit 104 as well as sends a response message to the mobile host 146. FIG. 28 (3) is an example of the address post message, which includes the home address α as well as the temporary address β or γ of the mobile host 146, a value of an autonomous flag F, and a broadcast address Bba, Cba on the network B, C. The autonomous flag F will be described later. FIG. 28 (4) is an example of the response message.

A mobile host transfer unit 104 stores the address post message into the home mobile host list hold unit

102, notifies the visitor migration communication control device 109 or 109' of the migration of the mobile host 146 by sending thereto a mobile host transfer message, and receives the data packet including the response. Further, according to a direction given by the packet transfer unit 103, the unit 104 transmits the mobile host transfer message both to the stationary host 151 and the device 109 or 109'. The unit 103 gives the direction when the value of the autonomous flag F is 1.

FIG. 32 (3) and FIG. 36 (5) are examples of the mobile host transfer message including the home address α, the temporary address β or γ, and the autonomous flag F. Since the mobile host transfer message is sent to the stationary host 151 is sent only when the autonomous flag F is 1; therefore, it does not necessarily include the value of the flag F. However, the identical message is sent both to the stationary host 151 and the visitor migration communication control device 109, 109' in this embodiment to simplify the construction of the mobile host transfer unit 104. FIG 32 (4) is an example of the response message.

As shown in FIG. 17, the home mobile host list hold unit 102 holds the home address α, the temporary address β, γ, the value of the autonomous flag F, and the broadcast address Bba, Cba on the network B, C, all of which are obtained from the mobile host transfer unit 104.

The packet monitoring unit 106 receives the packet destined for the home address α of the mobile host 146, then sends it to the packet transfer unit 103 when the stationary host 151 transmits the packet to the home address α of the mobile host 146 after the mobile host 146 has migrated to another network.

The packet transfer unit 103 has a payload including the noncapsulated data packet and the packet transfer message informing the transfer of the noncapsulated data packet, generates another data packet, and sends it to the temporary address β, γ of the mobile host 146. FIG. 32 (2) is an example of the packet transfer message. As is described the above, the packet transfer unit 103 directs the mobile host transfer unit 104 to transmit the mobile host transfer message to the stationary host 151 only when the autonomous flag in the home mobile host list hold unit 102 shows the value of 1. The operation conducted when the flag F is 1 will be described later.

When the stationary host 151 has problems in communicating with the mobile host 146 such as receiving the unusual mobile host transfer message, the address inquiry unit 105 is employed to solve the problems. That is, receiving from the stationary host 151 an address inquiry message, the address inquiry unit 105 transmits to the stationary host 151 a data packet which responds to the address inquiry by showing the address to be used in the communication. The address inquiry message includes a type field 132, a flag field 133, a sequence field 134, and a home address field 138, each of which having value 5, 1, a certain number, and α respectively; while the response message includes a temporary address field 139 filled with the temporary address β, γ as well as the flag field with 2, besides the type field 132, the sequence field 134, and the home address field 138 filled with the same values in the address inquiry message.

[visitor migration communication control device 109]

The visitor migration communication control device 109 implements the same function with the home migration communication control device 101. That is, when the stationary host 151 transmits an encapsulated data packet to the temporary address β of the mobile host 146, which is the updated address thereof since the mobile host has migrated to the network C, the visitor migration communication control device 109 transfers that encapsulated data packet from the temporary address β to temporary address γ. Then, the device 109 posts to the stationary host 151 the temporary address γ, so that the stationary host 151 will be able to communicate directly with the mobile host 146. However, whether or not the device 109 provides the above packet transfer service will be determined in accordance with a processing load put on the device 109 or with a initial setting given by a system operator; thus, the packet transfer service of the device 109 is not necessarily an obligation.

As shown in FIG. 18, the visitor migration communication control device 109 consists of the migration address unit 145 and the communication control unit 108. The migration address unit 145 further comprises the packet monitoring unit 106, a visitor mobile host list hold unit 110, a packet transfer unit 111, a mobile host transfer unit 112, a mobile host visit unit 113, and an autonomous support unit 114. The unit 106 and the unit 108 function the same as those in the home migration communication control device 101.

Receiving an autonomous packet transfer support check message inquiring if the visitor migration communication control device 109 provides the packet transfer service, the autonomous support unit 114 responds to it with the response message where the autonomous flag F shows 1 when the device 109 provides that service or 0 when it does not provide that service. FIG. 28 (1) is an example of the autonomous packet transfer support check message, while FIG. 28 (2) is an example of the response message including the autonomous flag F and the broadcast address Bba.

Receiving from the mobile host 146 the mobile host visit message which informs that the mobile host 146 has migrated to the network B, the mobile host unit 113 responds it with the response message after storing the mobile host visit message into the visitor mobile host list hold unit 110. The mobile host visit message includes the home address α and the temporary address β of the mobile host 146. FIG. 28 (5) is the format of the mobile host visit message, while the FIG. 28 (6) is the format of the response message.

Receiving from the mobile host transfer unit 104 in the device 101 the mobile transfer message informing that the mobile host 146 has migrated to the network C, the mobile host transfer unit 112 stores in the visitor mobile host list hold unit 110 the updated temporary address γ of the mobile host 146 and the value of the autonomous flag F by corresponding them to the home address α. The unit 112 also transmits to the stationary host 151 the mobile host transfer message in accordance with the direction from the packet transfer unit 111, as does the mobile host transfer unit 104 in the device 101.

As shown in FIG. 19, the visitor mobile host list hold unit 110 holds the home address α and the temporary address β on the network B, which are obtained from the mobile host 146 via the mobile host visit unit 113, as well as the temporary address γ and value on the autonomous flag F, which are obtained from the home migration communication control device 101 via the mobile host transfer unit 112.

The packet transfer unit 111, as dqes the packet transfer unit 103 in the home migration communication control device 101, transmits to the temporary address γ the data packet including the transfer message as well as orders the mobile host transfer unit 112 to transmit the mobile host transfer message.

[mobile host 146]

As shown in FIG. 20, the mobile host 146 includes the migration address unit 115, an address obtainment unit 116, the communication control unit 108, and an application processing unit 124 which mainly controls the communication of protocols located in higher layers including an application layer, such as TCP or layers located higher than it.

The migration address unit 115 comprises the a packet transmission unit 117, a transfer packet reception unit 118, an address hold unit 119, a migration unit 120, an autonomous support unit 121, an address post unit 122, a mobile host visit unit 123.

The migration address unit 115 comprising the above units is employed in transfer of data to the temporary address β or γ when the mobile host 146 migrates to the network B or C. Also receiving the data packet destined for the temporary address β or γ including the packet transfer message and the noncapsulated data packet, the device 115 transmits the noncapsulated data to the application processing unit 124.

In accordance with the order given by the application processing unit 124 when the mobile host migrates to the network B, C, the migration unit 120 controls the address obtainment unit 116, the autonomous support unit 121, the address post unit 122, the mobile host visit unit 123, and the address hold unit 119.

Directed by the migration processing unit 120, the address obtainment unit 116 obtains the temporary address β, γ of the mobile host 146 assigned when it migrates to the network B, C respectively. BOOTP in "Bill Croft and John Gilmore, BOOTSTRAP PROTOCOL RFC951, Sep., 1985" is an example of obtaining the temporary address; besides employing the BOOTP, the operator may input the temporary address β, γ assigned by a system administrator of the network B, C.

Directed by the migration unit 120, the autonomous support unit 121 sends the autonomous packet transfer support check message to inquire if the visitor migration communication control device 109, 109' attached to the networ:: B, C provides the packet transfer service and receives the response message to the inquiry. The autonomous packet transfer support check message is also sent to obtain the broadcast address Bba and Cba on the network B and C respectively.

Directed by the migration unit 120, the address post unit 122 sends the address post message to notify the home migration communication control device 101 of the temporary address β, γ. The address post message also informs whether or not the device 109, 109' provides the packet transfer service as well as the broadcast address Bba, Cba on the network B, C. If the response message from the visitor migration communication control device 109, 109' has the value 1 of the autonomous flag F, the mobile host visit unit 123 transmits to the visitor migration communication control device 109, 109' the mobile host visit message including the home address α as well as the temporary address β, γ respectively.

As shown in FIG. 21, the address hold unit 119 previously holds the home address α of the mobile host 146 and the broadcast address Aba on the network A. Now, the unit 119 newly holds the temporary address β or γ obtained from the address obtainment unit 116 via the migration unit 120 and the broadcast address Bba or Cba obtained from the autonomous support unit 121 via the migration unit 120.

When the mobile host 146 is attached to the network A and receiving a data packet destined for the home address α, the transfer packet reception unit 118 sends data etc. in the noncapsulated data packet to the application processing unit 124. On the other hand, when the mobile host 146 is attached to the network B and receiving a data packet destined for the temporary address β, the data packet including the packet transfer message and the noncapsulated data packet destined for α, the unit 118 sends to the application processing unit 124 data etc. in the noncapsulated data. Thus, the application processing unit 124 receives the data without being affected by the migration of the mobile across the networks.

Receiving the data to be transmitted and the instruction from the application processing unit 124, the packet transmission unit 117 generates a noncapsulated data packet whose destination address is the home address α and transmits it.

[stationary host 151]

As shown in FIG. 22, the stationary host 151 comprises the migration address unit 125 and the application processing unit 161 which mainly controls the communication of a protocol located in higher layers including application layer, such as TCP or layers located higher than the TCP and the communication control unit 108.

The migration address unit 125 comprises a transfer packet transmission unit 126, a packet reception unit 127, an address hold unit 128, an address inquiry unit 129, and the mobile host transfer unit 130.

The migration address unit 125 comprising the above units generates a noncapsulated data packet and sends it to the home address α when it is not notified that the mobile host 146 migrate to the network B or C and obtained the temporary address β or γ respectively. The unit 125 also generates an encapsulated data packet including as a payload the noncapsulated data packet and a data transfer message, which informs transfer of the noncapsulated data packet and sends it to the temporary address β, γ, when it is notified of the migration.

Receiving from the home migration communication control device 101 and the visitor migration communication control device 109, 109' the data packet including the mobile host transfer message which informs the migration of the mobile host 146, the mobile host transfer unit 130 stores into the address hold unit 128 the home address α and the temporary address β or γ of the mobile host 146 assigned on the network B or C respectively.

As shown in FIG. 23, the address hold unit 128 holds the home address α, the temporary address β or γ by corresponding them.

Directed by the application unit 161, the transfer packet transmission unit 123 generates a data packet destined for the home address α, and transmits it. However, if the address hold unit 128 holds the temporary address β or γ besides the home address α, the unit 126 generates an encapsulated data packet destined for the temporary address β or γ, which includes as a payload a noncapsulated data packet and a packet transfer message, which informs transfer of the noncapsulated data packet, and transmits it.

As is described the above, both the home migration communication control device 101 and the visitor migration communication control device 109, 109' generate the encapsulated data packet including the packet transfer message and the noncapsulated data and transmits it to the current temporary address of the mobile host 146. Owing to the device 101 or 109, 109', the stationary host 151 is able to transmit to the mobile host 146 both the noncapsulated data packet destined for the home address α and the encapsulated data packet destined for the temporary address β or γ without failure even when the address hold unit 128 fails to hold the current temporary address β or γ and the stationary host 151 transmits the data packet to the outdated address of the mobile host 146.

The packet reception unit 127 receives a data packet which is sent from the mobile host 146 and has Sa as its destination address, and sends the data etc. in it to the application unit 161.

When the address inquiry unit 129 has problems such as that it received an illegal mobile host transfer message or that it cannot communicate with the mobile host 146 successfully, it transmits a data packet including an address inquiry message in order to inquire of the host migration communication control device 101 the address which is currently used to communicate with the mobile host 146.

[construction of data packet]

As shown in FIG. 24 (a), (b), (c), there are three kinds of data packets, each data packet 210, 220, 230, includes each of header 211, 221, 231 and payload 212, 222, 232 respectively.

The header 211 of the data packet 210 includes a destination address 201, and a source address 202. Also the payload 212 consists of a transmission data 203.

The header 221 of the data packet 220 includes the destination address 201 and the source address 202. Also the payload 222 consists of a message 204.

The header 231 of the data packet 230 includes the destination address 201 and the source address 202. Also the payload 232 consists of the message 204, which is employed as the packet transfer message, and a noncapsulated data packet 210. Also each header 211, 221, 231 includes information showing presence or absence of the message 204 as a protocol number etc.

The message 204 includes some of the fields in FIG. 25 in accordance with its type.

The type of the message 204 is indicated in the message type field 132. Besides the above types, the message 204 is also employed as an echo message for examining whether or not a host employs an appropriate operation in accordance with the message.

A flag field 133 indicates whether or not the message 204 is a response. When the message 204 is not the response, the field 133 further indicates whether or not the message 204 requests a response.

A sequence field 134 gives a single number both to the request message and its response message, thereby the request message and the response message are corresponded.

An autonomous flag field 135 contains a value of the autonomous flag F indicating whether or not the visitor migration communication control device 109,109' provide the packet transfer service.

A counter field 136 contains a counter indicating the number of the visitor migration communication control devices employed to transfer the encapsulated data packet consisting of the packet transfer message and the noncapsulated data packet. The visitor migration communication control device increments the counter in the received message packet by 1, and gives it to the message to be transmitted. When the incremented number is greater than the predetermined number, the received message packet is disposed. ,

A status field 137 of the response message indicates presence or absence of an error in a transmission/reception of the data packet. For example, it indicates an error in authentication information, which will be described later, or the address inquiry message which cannot or should not be responded.

A home address field 138, a temporary address field 139, and a broadcast address field 140 indicates the home address as well as the temporary address of the mobile host 146 or the broadcast address on its home network or on the network it migrates. However, what the broadcast address field 140 indicates depends on type of the message 204. Whether the message 204 is the request or the response also devices the content of the broadcast address field 140.

The authentication information field 141 indicates if a source address coincides with the sender's address.

[outline of communication operation]

The home migration communication control device 101 and the visitor migration communication control device 109,109' is basically employed to transfer the data packet transmitted by the stationary host 151 as well as post to the stationary host 151 the updated temporary address of the mobile host 146. Understanding of such operations will be helped by the following two points.

1. Transfer of the data packet and posting of the updated temporary address are conducted only when the mobile host 146 migrates from its home network to another network. The home network refers to the one to which the home migration communication control device is attached.

2. Posting of the updated temporary address is conducted only when the autonomous flag F is 1, which indicates the visitor migration communication control device 109, attached to the same network as is the mobile host 146, provides the packet transfer service. Otherwise, the data packet transmitted by the stationary host 151 to the posted temporary address will not be received by the mobile host 146 when the mobile host 146 migrates to another network.

[communication operation 1]

An example of the communication operation is described hereunder. In the communication operation 1 the visitor migration communication control device 109,109' provides the packet transfer service when the mobile host 146 migrates from the network A to the network B, further from the network B to the network C.

[migration from network A to network B]

The operation at the migration of the mobile host 146 from the network A to the network B is described with referring to FIGs. 26-29. FIG. 26 shows a flow of the data packet transmitted between the devices; FIG. 27 shows a communication sequence of the data packet; FIG. 28 shows construction of each data packet; and FIG. 29 shows the content of the address hold unit 119 etc.

When the mobile host 146 is attached to the network A, the home mobile host list hold unit 102 in the home migration communication control device 101 holds the home address α both as the home address and the temporary address of the mobile host 146. Thereby the home migration communication control device 101 detects that the mobile host 146 is attached to the network A.

The address hold unit 119 in the mobile host 146 holds the home address α and the broadcast address Aba on the network A.

When the mobile host 146 migrates to the network B, the application unit 124 orders the operation of the migration unit 120 in accordance with the instruction given by the operator. The temporary address β is assigned to the mobile host 146 on the network B, and the address obtainment unit 116 obtains it. The migration unit 120 stores into the address hold unit 119 the temporary address β together with the home address α and the broadcast address Aba.

(1) The autonomous support unit 121 transmits to the visitor migration communication control device 109, which is attached to the network B, the data packet including the autonomous packet transfer support check message 147 which holds the home address α and the temporary address β. The destination address of the data packet is the broadcast address shared by every network, such as an address where every bit is 1. The message 147 does not necessarily hold the home address α and the temporary address β although

they can be used in checking the security of the network if it does. Also the message 147 holding the home address α and the temporary address γ can take the place of a mobile host visit message 146, which will be described later.

(2) The autonomous support unit 114 in the visitor migration communication control device 109 responds to the autonomous support unit 121 with the response message 147R where broadcast address Bba is set and the autonomous flag F in the autonomous flag field 135 indicates 1 to inform that the device 109 provides the packet transfer service.

The mobile host 146 transmits the data packet to the visitor migration communication control device 109. The broadcast address Bba is employed as the destination address of the data packet and it is set in the response message 147R; however, this is not an obligation.

That is, when the response message 147R does not hold the broadcast address Bba, the following means can be employed. First, the broadcast address shared by every network can be employed, which is described in the above. Second, the source address, which is set in the header of the data packet comprising the response message 147R, can be employed. Third, a so called name service can be employed, where a server device on the network system informs the broadcast address Bba. Finally, when the address assigned to each of the devices, which are attached to the network, consists of the network address being unique for the network and a device address being unique for the devices, and the broadcast address on each network consists of such network address and the device address where the value of every bit is 1, the network address Bba can be generated by employing the network address included in the temporary address β of the mobile host 146.

(3) The address post unit 122 transmits to the home migration communication control device 101 the address post message 148. The message 148 includes the value 1 of the autonomous flag F, which is obtained from the response message, home address α, the temporary address β on the network B, and the broadcast address Bba, and the broadcast address Aba is the destination address of the address post message 148.

When the address post unit 107 in the home migration communication control device 101 receives the address post message 148, the mobile host transfer unit 104 stores in the home mobile host list hold unit 102 the temporary address β, the value 1 of the autonomous flag 1, and the broadcast address Bba by corresponding them to the home address α. Since the home address α had been stored as the temporary address before the temporary address β was stored, the mobile host transfer unit 104 knows that the mobile host 146 has migrated from the network A to the network B; therefore, it does not transmit the mobile host transfer message to the visitor migration communication control device 109,109'. That is, the data packet transmitted by the stationary host 151 to the home address α of the mobile host 146 is received by the home migration communication control device 101 and transferred thereby to the temporary address β; therefore, the visitor migration communication control device 109,109' is not employed here.

(4) The address post unit 107 notifies the address post unit 122 that it has received the address post message 148 by sending the response message 148R.

(5) Since the visitor migration communication control device 109 provides the packet transfer service, the mobile host visit unit 123 transmits to the visitor migration communication control device 109 the mobile host visit message 149 including the home address α and the temporary address β, so that the device 109 is notified that the mobile host 146 has migrated to the network B. The mobile host visit message 149 is destined for the broadcast address Bba.

The mobile host visit unit 113 in the visitor migration communication control device 109 receives the mobile host visit message 149 and stores into the visitor mobile host list hold unit 110 the home address α as well as the temporary address β. The temporary address β is stored also as the updated temporary address of the mobile host 146, which will be assigned when the mobile host 146 migrates from the network B to another network; thereby, the visitor migration communication control device 109 detects that the mobile host is currently attached to the network B.

(6) The mobile host visit unit 113 notifies the mobile host visit unit 123 by sending the response message 149R that it has received the mobile host visit message 149.

[communication between the stationary host 151 and the mobile host 146 on the network B]

The operation at the communication between the stationary host 151 and the mobile host 146 when the mobile host is attached to the network B is described hereunder with referring to FIGs. 30-33, which are relevant for FIGs.26-29.

(1) The application unit 161 in the stationary host 151 directs the transmission of the noncapsulated data packet, whose destination is the home address α, despite the migration of the mobile host 146. Immediately after the mobile host 146

migrates to the network B, that is, when the address hold unit 128 does not hold the home address α and the temporary address β, the transfer packet transmission unit 126 is not notified of the migration; therefore, it generates the noncapsulated data packet 152 and transmits it to the home address α in accordance with the direction from the application unit 151.

The noncapsulated data packet 152 is not received by the mobile host 146, which is not attached to the network A, but by the packet monitoring unit 106 in the home migration communication control device 101 since the home mobile host list hold unit 102 in the device 101 holds the home address α as well as the temporary address β, which coincides with the destination address of the noncapsulated data packet 152.

(2) The packet transfer unit 103 in the home migration communication control device 101 generates an encapsulated data packet including the noncapsulated data packet 152, which is received by the packet monitoring unit 106, and the packet transfer message 153, which informs the transfer of the noncapsulated data packet 152; and transmits it to the temporary address β. The packet transfer message 153 includes the value 0 in the field 133, which indicates that no response is requested, as well as the value 0 on the counter in the field 136, which indicates that the packet transfer message is the first message added to the noncapsulated data packet 152. As is described, no response is requested by the packet transfer message 153. That is, the application unit 161 of the stationary host 151 and the application unit of the mobile host 146, rather than the home migration communication control device 101 and the migration address unit 115, confirm that the mobile host 146 receives the noncapsulated data packet 152.

The transfer packet reception unit 118 in the mobile host 146 receives the encapsulated data packet including the packet transfer message 153 and the noncapsulated data packet 152, since it is destined for the temporary address β, which is held in the address hold unit 119. The unit 118 then detects that the destination address of the noncapsulated data packet 152 is the home address α, and sends the data etc. in the noncapsulated data packet 152 to the application unit 124.

Thus, the communication between the application unit 124 and the application unit 161 is not affected by the migration of the mobile host 146.

(3) The packet transfer unit 103 transmits the encapsulated data packet including the data packet transfer message. It also directs, after detecting that the autonomous flag F indicates 1, the mobile host transfer unit 104 to transmit to the stationary host 151 the data packet including the mobile host transfer message 154 where the home address α and the temporary address β are set. Finally, the unit 104 transmits the data packet to the stationary host 151.

The mobile host transfer unit 130 in the stationary host 151 receives the mobile host transfer message and stores into the address hold unit 128 the home address α and the temporary address β.

(4) The mobile host transfer unit 130 responds to the mobile host transfer unit 104 with the response message 154R.

(5) When the application unit 161 directs the transmission of the noncapsulated data packet to the home address α after the address hold unit 128 holds the home address α and the temporary address β, the transfer packet transmission unit 126 first generates a noncapsulated data packet destined for the home address α, then generates an encapsulated data packet including it and a packet transfer message 155. The encapsulated data packet is then transmitted to the temporary address β. Thus, once the home migration communication control device 101 notifies the stationary host 151 of the home address α and the temporary address β, the stationary host 151 is able to transmit the data packet to the temporary address β of the mobile host 146, and the home migration communication control device 101 is not employed.

On the other hand, when data is transmitted from the mobile host 146 to the stationary host 151, the Sa is employed as the destination address α and the home address is employed as the source address; and the noncapsulated data packet is transmitted from the address α to the address Sa.

Thus, even when all the noncapsulated data transmitted by the stationary host 151 is destined for the home address α, the home migration communication device 101 transfers the data to the updated temporary address of the mobile host; thereby, the communication between the mobile host 146 and the stationary host 151 is implemented, and the conventional device can be employed as the stationary host 151, which broadens a practicability of the network system.

Whereas, when the network system checks the original source address of the data packet or a transfer path of the data packet, the transmission unit may be built in the mobile host 146 like the transfer packet transmission unit 126 in the stationary host 151, and also the reception unit may be built in the stationary host 151 like the transfer packet reception unit 118 in the mobile host 146; and the encapsulated data packet including the packet transfer message and the noncapsulated data packet may be transmitted therebetween.

[migration from network B to network C]

The operation at the migration of the mobile host 146 from the network B to the network C is described hereunder with referring to FIGs. 34-37, relevant for FIGs. 26-29.

(1)-(4) The operation related to transmission of an autonomous packet transfer support check message 147', a response message 147R', an address post message 148', and a response message 148' between the mobile host 146 and the visitor migration communication control device 109' is substantially same as the operation related to transmission of messages between the mobile host 146 and the visitor migration communication control device 109, which is conducted when the mobile host 146 migrates to the network B. However, the operation at the migration from the network A to the network B and the operation at the migration from the network B and the network C are different from each other in part of the operation of the home migration communication control device 101 conducted after it responds to the received address post message 148' with the response message 148R.

(5) When the address post unit 107 receives the address post message 148', the mobile host transfer unit 104 in the home migration communication control device 101 detects that the mobile host been attached to the network B before migrating to the network C since the temporary address β has been stored as the temporary address. Then, the mobile host transfer unit 104 sends to the visitor migration communication control device 109 the data packet including both the home address α and the temporary address γ, so that the device 109 transfers the data packet transmitted by the statinnary host 151 from the temporary address β to the temporary address γ. The data packet received by the visitor migration communication control device is destined for the broadcast address Bba.

In accordance with the address post message 148', the mobile host transfer unit 104 stores into the home move host list hold unit 102 the temporary address γ, the value 1 of the autonomous flag F, and the broadcast address Cba by corresponding them to the home address α.

Receiving the data packet including the mobile host transfer message 150, the mobile host transfer unit 112 in the visitor migration communication control device 109 stores into the visitor mobile host list hold unit 110 the temporary address γ newly assigned to the mobile host 146 and the value 1 of the autonomous flag F by corresponding them to the home address α.

(6) The mobile host transfer unit 112 notifies the mobile host transfer unit 104 that it has received the mobile host transfer message 150 by sending thereto the response message 150R.

(7), (8) The transmission of a mobile host visit message 149' and a response message 149R' between the mobile host 146 and the visitor migration communication control device 109', which is conducted when the device 109' provides the packet transfer service, is substantially same as the transmission of messages between the mobile host 146 and the visitor migration communication control device 109, which is conducted when the mobile host 146 migrates to the network B.

[communication between mobile host 146 attached to network C and stationary host 151]

Transmission of the data packet from the stationary host 151 to the mobile host 146 when the mobile host is attached to the network C is described with referring to FIG. 38-41, which are relevant for FIG. 26-29.

The transmission is substantially same as the transmission between the stationary host 151 and the mobile host 146 when the mobile host 146 is attached to the network B, except that the visitor migration communication control device 109 instead of the home migration communication control device 101 is employed.

(1) When the stationary host 151 is not notified that the mobile host 146 has migrated from the network B to the network C, the stationary host 151 generates the encapsulated data packet including the noncapsulated data packet, which is destined for the home address α, and the packet transfer message 156; then transmits it to the temporary address β. This is substantially the same as (5) in the communication between the stationary host 151 and the mobile host 146 attached the network B.

The data packet transmitted by the stationary host is not received by the mobile host 146 since the mobile host is not attached to the network B. The data packet is received by the packet monitoring unit 106 in the visitor migration communication control device 109 since the visitor mobile host list hold list unit thereof holds the temporary address β besides the temporary address γ.

(2) The visitor migration communication control device 109 transmits to the temporary address γ of the mobile host 146 the data packet including the packet transfer message 157, which is substantially same as (2) in the communication between the stationary host 151 and the mobile host 146 on the network B except a difference described hereunder.

The home mobile host migration communication control device 101 receives the noncapsulated data packet 152 and generates an encapsulated data packet comprising the received noncapsulated data packet 152 and the packet transfer message 153. On the other hand, the visitor migration communication control device 109 receives the encapsulated data packet comprising the packet transfer message 156 and the packet transfer unit 111 converts the data packet by changing the destination address from the temporary address β into the temporary address γ as well as converting the packet transfer message 156 into the packet transfer message 157, whose value on the counter is incremented by 1.

(3)-(5) The visitor migration communication control device 109, the stationary host 151, and the mobile host 146 on the network C operate substantially same as the home migration communication control device 101, the stationary host 151, and the mobile host 146 on the network B, which is described the above in (3)-(5); thereby the mobile host transfer message 158 and the response message 158R are transmitted, and the data packet including the packet transfer message 160 is transmitted by the stationary host 151 to the mobile host 146 attached to the network C.

If the stationary host 151 does not transmit any data packet to the mobile host 146, which is attached to the network B, the stationary host is not notified of either the temporary address β or the temporary address γ; therefore, the stationary host 151 transmits the data packet to the home address α even when the mobile host 146 has migrated from the network B to the network C. When this occurs, the home migration communication control device 101, as does the visitor migration communication device 109, transfers the data packet from the home address α to the temporary address γ; then notifies the stationary host 151 of the updated temporary address γ of the mobile host 146 so that the stationary host 151 will be able to directly transmit the data packet, which comprises the packet transfer message, to the mobile host 146 attached to the network C.

Further, when the mobile host 146 migrates to the network, to which the visitor migration communication control device is attached to provide the packet transfer service, the stationary host 151 may transmit the data packet destined for any of the addresses α, β, or γ. When the data packet is transmitted to the home address α or the temporary address γ, the home migration communication control device 101 or the visitor migration communication control device 109', which is notified of the updated temporary address of the mobile host 146, transfers the data packet to the updated temporary address; then it notifies the stationary host 151 of the updated temporary address of the mobile host.

When the data packet is transmitted to the temporary address β of the mobile host 146, the visitor migration communication control device 109 receives it. Since the device 109 is notified of only the temporary address γ, it transmits the data packet comprising the packet transfer message to the temporary address γ as well as transmits the mobile host transfer message to notify the stationary host 151 of the temporary address γ. The visitor migration communication control device 109' receives the data packet comprising the packet transfer message, which is destined for the temporary address γ, and transmits it to the updated temporary address of the mobile host 146; then transmits the mobile host transfer message to notify the stationary host 151 of the updated temporary address. Also the visitor migration communication control device 109' obtains the address of the visitor migration communication control device 109 from the source address of data packet transmitted thereby, and transmits the mobile host transfer message to the device 109. Thus, the visitor migration communication control device 109' obtains the updated temporary address of the mobile host 146, and transfers the data packet to the mobile host 146 as well as notifies stationary host 151 of the obtained updated temporary address.

[communication operation 2]

Another example of the communication operation is described hereunder. In the communication operation 2 the visitor migration communication control device 109 does not provide the packet transfer service when the mobile host 146 migrates from the network A to the network B, further from the network B to the network C.

As shown in FIG. 42, when the device 109 does not provide the packet transfer service, the autonomous packet transfer support check message 181, transmitted by the mobile host 146 which has migrated from the network A to the network B, is responded with the response message 181R where the autonomous flag F in the autonomous flag field 135 indicates 0. Thereby, the autonomous flag field 135 in the address post message 182, which is transmitted by the mobile host 146 to the home migration communication control device 101, obtains the value 0, and the value 0 is held in the home mobile host list hold unit 102 in the device 101. The mobile host 146 does not transmit the mobile host visit message to the visitor migration communication control device 109.

As shown in FIG. 43, receiving from the stationary host 151 the noncapsulated data packet 183, which is destined for the home address α, the home migration communication control device generates the encapsulated data packet comprising the received noncapsulated data packet 183 and the packet transfer message 184, and transmits it to the temporary address β, as is in the communication operation 1.

However, recognizing the value 0 on the autonomous flag F, which is held in the home mobile host list hold unit 102, the device 101 does not transmit to the stationary host 151 the mobile host transfer message including the temporary address β. Therefore, every data packet transmitted by the stationary host 151 is destined for the home address α, and it is transferred to the mobile host 146 by the home migration communication control device 101. Thus, the stationary host 151 is not notified of the temporary address β since the data packet transmitted to the address other than the home address α is not transferred by the device 109; therefore it is not received by the mobile host 146 when it departs the network B to migrate to the network C.

When the visitor migration communication control device 109', which is attached to the network, provides the packet transfer service, the home migration communication control device 101 notifies the stationary host 151 of the temporary address γ when it transmits the noncapsulated data to the home address α, so that the stationary host 151 is able to directly transmit the data packet comprising the noncapsulated data packet and the packet transfer message to the mobile host 146 on the network C.

When the visitor migration communication control device 109 does not provide the packet transfer service, the home migration communication control device 101 does not necessarily notify the device 109 of the temporary address γ of the mobile host 146 assigned when it has migrated from the network B to the network C. However, the construction of the device 101 will be simplified if it conducts the same operation either or not the packet transfer service is provided since the visitor migration communication control device 109 ignores the mobile host transfer message.

Also the device 109 may respond to the autonomous packet transfer support check message 181 only when it provides the data packet transfer service; therefore, the presence or absence of the response message 181R indicates to the mobile host 146 whether or not the data packet transfer service is provided. In the above operation the value 0 of the autonomous F also indicates that the packet transfer service is not provided, whereas absence of the response message to the message 181 can indicate the absence of the packet transfer service, which will simplify construction of mobile host 146.

[communication operation 3]

The final example of the communication operation is described hereunder. In the communication operation 3 the visitor migration communication control device 109' does not provide the packet transfer service while the visitor migration communication control device 109 does.

As shown in FIG. 44, when the packet transfer service is not provided by the visitor migration communication control device 109', the mobile host 146 transmits to the home migration communication control device 101 the address post message 182' where the value 0 is set at the autonomous flag F. Then, the home migration communication control device 101 transmits to the device 109 the mobile host transfer message 185 by setting the value 0 at the autonomous flag F.

When detecting the value 0 at the autonomous flag F, the visitor migration communication control device 109 ceases to provide the packet transfer service.

As shown in FIG. 45, even after cease of the data packet transfer service, the stationary host 151 may transmit to the temporary address the data packet comprising the noncapsulated data packet and the packet transfer message 186.

When this happens, the visitor migration communication control device 109 obtains the noncapsulated data packet 187 from the received encapsulated data packet and transmits it to its destination address, the home address α. The noncapsulated data packet 187 is then received by the home migration communication control device 101, which is attached to the network A. Finally, the home migration communication control device 101 transfers the noncapsulated data packet 187 together with the packet transfer message 188 to the temporary address γ of mobile host 146, which is attached to the network C.

The visitor migration communication control device 109 notifies the stationary host 151 that the mobile host 146 is attached to the network A instead of the network C by sending the mobile host transfer message 189 where the home address α is set in the temporary address field 139. Then, the stationary host 151 transmits the noncapsulated data packet 187 to the home address α, and it is transferred by the home migration communication control device 101, which is employed to take the place of the visitor migration communication control device 109. As another option, the device 109 may send the mobile host transfer message 189 where the invalid address is set, such as the address where every bit is 1. Then, the home migration communication control device 101 may notify the stationary host 151 of the home address α in accordance with the address inquiry obtained from the stationary host 151.

The operation described the above will be employed when the visitor migration communication control device 109 ceases to provide the packet transfer service operation regardless whether or not the device 109' provides the packet transfer service.

On the other hand, the visitor migration communication device 109 may restart the packet transfer service even when the device 109' ceases to provide the service.

In this case, the home migration communication control device 101 needs to provide the visitor migration communication control device 109 with the updated temporary address at every migration of the mobile host 146 unless the mobile host migrates to the network to which another visitor migration communication control device is attached and provides the packet transfer service. To realized it, for example, when the value of the autonomous flag F in the address post message is 0 to indicate that the device 109' does not provide the packet transfer service, the broadcast address Bba as the destination address of the mobile host transfer message, which is transmitted to the device 109, will not be renewed.

Additionally, the broadcast address as the destination address of the data packet, which is transmitted by the mobile host 146, can be replaced with the address Ha, Va, Va', each of which is unique to each device. The address unique to each device will be obtained by detecting the source address of the data packet received from each device, or by employing a so called name service.

Also in the second embodiment, the home migration communication control device 101 detects whether or not the mobile host 146 is attached to the same network from what is held as the temporary address in the address hold unit; to be precise, whether or not the home address α is held as the temporary address. However, this can also be detected by knowing in which table the temporary address is held. For example, when the device 101 and the mobile host 146 are attached to the same network, the first table holds the addresses, such as the home address α; whereas, the second table holds the addresses when the device 101 and the mobile host 146 are attached to the different network from each other. Value of the autonomous flag F, 0 or 1, can also be utilized in the same way.

Further, the home migration communication control device 101 and the visitor migration communication control device 109, 109' may be employed as a host such as the mobile host 146 or the stationary host 151.

Finally, the home migration communication control device 101, the visitor migration communication control device 109, the mobile host 146, and the stationary host 156 may be constructed identically and can be replaced with each other.

Although in the embodiment the application unit 124 starts its operation before being notified of updated temporary address β; therefore it always transmits the data packet to the home address α of the mobile host 146, it can transmit the data to the temporary address β if is starts its operation after obtaining the temporary address β.

Although the present invention has been fully described by way of examples with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. Therefore, unless otherwise such changes and modifications depart from the scope of the present invention, they should be constructed as being included therein.

Claims

1. A migration communication control device constructed to control a communication between a mobile node and a partner node, the mobile node migrating across networks and obtaining an address assigned on each network while the partner node being a communication partner of the mobile node, comprising a first migration control unit, a second migration control unit, a third migration control unit, the second migration control unit being placed on the mobile node and the third migration control unit being placed on the partner node,

    wherein the first migration control unit comprises:

    packet transfer means for receiving a packet which was destined for an outdated address of the mobile node, the outdated address assigned when the mobile node migrated to a network to which the first migration control unit is attached, generating a conversion packet which holds an updated address instead of the outdated address, and transmitting the conversion packet; and

    address post means for transmitting an address post message which indicates the updated address of the mobile node to the third migration control unit, the third migration control unit transmitting the packet received by the packet transfer means, and

    the second migration control unit comprises:

    migration post means for transmitting to the first migration control unit a migration post message which indicates the updated address of the mobile node when the mobile node migrates to another network; and

    packet resumption means for receiving the conversion packet from both the first migration control unit and the third migration control unit and resuming an original packet from the conversion packet, and

    the third migration control unit comprises:

    packet conversion means for converting a destination address of a packet, the packet to be transmitted to the mobile node, into the updated address indicated by the address post message, the address post message sent by the first migration control unit, and transmitting it to the mobile node.

2. The migration communication control device of Claim 1, wherein the migration post means in the second migration control unit transmits an identification key included in the migration post message, the identification key being employed to identify the mobile node.

3. The migration communication control device of Claim 2, wherein the identification key is an address of the mobile node assigned at one network before the network to which the mobile node is currently attached.

4. The migration communication control device of Claim 2, wherein the identification key is an address of the mobile node assigned before its initial migration.

5. The migration communication control device of Claim 1; wherein the second migration control unit is constructed to transmit to the third migration control unit the packet which has the same format as the resumed packet.

6. The migration communication control device of Claim 1, wherein the first migration control unit further comprises:

address hold means for holding the outdated address and the updated address by corresponding them with each other; and

address comparison means for comparing the destination address of the received packet with the outdated address, wherein

the packet transfer means generates the conversion packet and transmits it when the address comparison means detects that the destination address of the received packet coincides with the outdated address.

7. The migration communication control device of Claim 1, wherein the first migration control unit further comprises:

address hold means for holding the outdated address and the updated address by corresponding them with each other; and

address comparison means for comparing the destination address of the packet received by the packet transfer means with the outdated address, wherein

the address post means transmits the address post message which indicates the updated address of the mobile node to the third migration control unit, the third migration control unit transmitting the packet received by the packet transfer means, when the address comparison means detects that the destination address of the packet coincides with the outdated address.

8. The migration communication control device of Claim 1, wherein the second migration control unit further comprises:

address hold means for holding the outdated address and the updated address by corresponding them with each other; and

address comparison means for comparing the updated address with the destination address of the packet received from one of the first migration control unit and the third migration control unit, wherein

the packet resumption means resumes the original packet from the conversion packet when the address comparison means detects that the updated address coincides with the destination address of the packet received from one of the first migration control unit and the third migration control unit.

9. The migration communication control device of Claim 1, wherein the third migration control unit further comprises:

address hold means for holding the outdated address and the updated address of the mobile node by corresponding them with each other; and

address comparison means for comparing the outdated address in the address hold means with the destination address of the packet to be transmitted to the mobile node, wherein

the packet conversion means converts the destination address of the packet to be transmitted to the mobile node into the updated address which corresponds to the outdated address in the address hold means when the address comparison means detects the outdated address in the address hold means coincides with the destination address of the packet.

10. The migration communication control device of Claim 1, wherein there are a plurality of the first migration control units, and the second migration control unit transmits the migration post message to at least one of the first migration control units.

11. The migration communication control device of Claim 10, wherein the migration post means in the second migration control unit transmits the migration post message to the first migration control unit which is attached to the network to which the mobile node was attached before its migration,

each of the first migration control units has migration post means for transmitting to one of the other first migration control units a migration post message to post the same address as the updated address indicated by the migration post message received from the second migration control unit, and

each of the first migration control units has migration post means for transmitting a migration post message from one of the other first migration control units to another first migration control unit to post the same address as the updated address indicated by the received migration post message.

12. The migration communication control device of Claim 11, wherein each of the first migration control units and the second migration control unit further comprise pointer hold means for holding pointers related to the first migration control unit to which the migration post message is transmitted, and wherein

the migration post means in each of the first migration control units and the migration post means in the second migration control unit transmit the migration post message to each of the addresses related to each of the pointers.

13. The migration communication control device of Claim 12, wherein each of the pointers is a broadcast address of the network to which one of the first migration control units is attached.

14. The migration communication control device of Claim 12, wherein each of the pointers is an address which is assigned to one of the first migration control units uniquely.

15. The migration communication control device of claim 12, wherein each of the pointers is the address of the mobile node which is assigned when the mobile node is attached to the same network as is the first migration control unit, and

the migration post means in the first migration control unit and the migration post means in the second migration control unit obtain the broadcast address of the network to which each of the first migration control units is attached with referring to the address of the mobile node, and transmits the migration post message to the obtained broadcast address.

16. The migration communication control device of Claim 12, wherein the pointer hold means in the second migration control unit holds a pointer related to a first migration control unit for the latest migration, which is the first migration control unit being attached to one network before the network to which the mobile node is currently attached, and

the pointer hold means in the first migration control unit holds a pointer related to another first migration control unit attached to the same network as was the mobile node attached before migrating to the network to which the first migration control unit is attached.

17. The migration communication control device of Claim 12, wherein the second migration control unit further transmits to the first migration control unit the pointer by sending thereto the migration post message, the pointer to be held by the first migration control unit.

18. The migration communication control device of Claim 17, wherein the first migration control unit stores into the pointer hold means the pointer when it receives from the second migration control unit the migration post message by corresponding the pointer with the updated address indicated by the received migration post message.

19. The migration communication control device of Claim 11, wherein each of the first migration control units further comprises:

address hold means for holding the outdated address and the updated address by corresponding them with each other, wherein

migration post message means stores into the address hold means the outdated address and the updated address by corresponding them with each other when it receives from the second migration control unit the migration post message, while converts the updated address in the address hold means into the updated address indicated by the migration post message when it receives from the first migration control unit the migration post message and the outdated address indicated by the migration post message coincides with one of the updated addresses in the address hold means.

20. The migration communication control device of Claim 1, wherein the first migration control unit is placed on a gateway, which connects networks.

21. The migration communication control device of Claim 1, wherein the first migration control unit is placed on the network as an individual node.

22. The migration communication control device of Claim 10, wherein the migration post means in the second migration control unit transmits the migration post message to a home migration control unit, the home migration control unit being the first migration control unit which is attached to a network where the mobile node left for its initial migration, and

the home migration control unit further

comprises home migration post means for transmitting a migration post message to a first migration control unit for the latest migration, the first migration control unit for the latest migration being the first migration control unit which is attached to the network where the mobile node left for the latest migration, to post the same updated address as is indicated by the migration post message received from the second migration control unit.

23. The migration communication control device of Claim 22, wherein the first migration control unit further comprises migration post means for transmitting the migration post message indicating the updated address of the mobile node to one of the other first migration control units when the conversion packet destined for the outdated address of the mobile node was sent therefrom to the first migration control unit.

24. The migration communication control device of Claim 22, wherein the migration post means in the second migration control unit transmits to the home migration control unit the migration post message where a home address and the updated address are corresponded with each other, the home address assigned when the mobile node is attached to the same network as is the home migration control unit,

and each of the packet transfer means and the address post means in the home migration control unit transmits the conversion packet and the address post message respectively with referring to the above home address and the updated address.

25. The migration communication control device of Claim 24, wherein the second migration control unit further comprises an outdated address post means for transmitting to the first migration control unit for the latest migration an outdated address post message where the outdated address and the home address are corresponded with each other, the outdated address being assigned to the mobile node before the latest migration,

the home migration post means in the home migration control unit transmits to the said first migration control unit for the latest migration the migration post message where the above home address and the updated address are corresponded with each other, and

the packet transfer means and the address post means in the first migration control unit for the latest migration transmit the conversion packet and the address post message respectively in accordance with the outdated address and the updated address, the outdated address and the updated address being corresponded with each other via the home address.

26. The migration communication control device of the Claim 25, wherein the outdated address post means in the second migration control unit transmits the above outdated address post message at a migration of the mobile node preceding the latest migration, and

each of the migration post means in the second migration control unit and the home migration post means in the home migration control unit transmits the above migration post message at the latest migration of the mobile node.

27. The migration communication control device of Claim 22, wherein the second migration control unit further comprises home migration control unit pointer hold means for holding a pointer related to the home migration control unit,

the migration post means in the second migration control unit transmits the migration post message to the address related to the pointer,

the home migration control unit further comprises pointer hold means for the latest migration for holding a pointer related to the first migration control unit for the latest migration, and

the home migration post means in the home migration control unit transmits the migration post message to the address related to the pointer.

28. The migration communication control device of Claim 27, wherein each of the above pointers is the broadcast address of the network to which each of the first migration control units is attached.

29. The migration communication control device of Claim 27, wherein each of the above pointers is the address assigned to each of the first migration control units uniquely.

30. The migration communication control device of Claim 27, wherein the second migration control unit further comprises pointer obtainment means for requesting to the first migration control unit for the latest migration the pointer related to the first migration control unit for the latest migration, and

the migration post means in the second migration control unit posts the obtained pointer to the home migration control unit together with the updated address by sending thereto the migration post message.

31. The migration communication control device of Claim 30, wherein the migration post means in the second migration control unit posts to the